HITS 算法:

核心概念：1）权威网页（Authority）：指某个领域或某个话题相关的高质量网页　2）中心网页（Hub）：类似中介，指向了很多高质量的权威网页　　HITS 目的：在海量网页中找到并区分这些与用户查询主题相关的高质量“Authority”与“Hub”网页，尤其是“Authority”（因为更能满足用户的信息需求）

$$\forall p, a(p) = \sum_{i=1}^{n} h(i) \qquad \forall p, h(p) = \sum_{i=1}^{n} a(i)$$

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：

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

或者，可采用如下迭代式：

- $a_{k+1} = (M^T M)^k M^T a_0, \quad h_{k+1} = (MM^T)^{k+1} h_0$
- 其中，a_0, h_0为Authority/Hub向量的初始值，可设为全1向量

每次迭代都要归一化！

$a_{k+1} = M^T M a_k, h_{k+1} = MM^T h_k$。

解特征方程（A-λI）x=0 且$\sum x_i^2=1$得概率 x。

优缺点：相比于 PageRank，HITS 算法是一种能够区分网页功能的排序算法　优点：1）更好地描述互联网组合特点 2）主题相关，因此可以单独用于网页排序　缺点：1）需要在线计算，时间代价较大 2）对链接结构变化敏感，且依然可能受到“链接作弊”的影响 3）初始页面的选择对查询结果有影响，需要找到好的选择算法

个性化检索

基于内容的推荐技术 (用户的偏好一般相对稳定)：物品画像&用户画像(基于用户评分进行加权)采用余弦相似性度量进行评分　优缺点：优点：1）每个人的推荐过程相互独立，不需要其他用户的数据 2）可以为具有独特偏好的用户进行有效推荐 3）可以推荐新项目或非热门项目 推荐结果有着较好的可解释性　缺点：1）找到合适的特征是一件困难的事 2）给新用户推荐项目，永远是一个困难的任务 3）过度特化（Overspecialization）现象

多样化评估：最大边界相关性 MMR　　基于路径推荐：知识图谱代替向量化画像，基于图谱上的游走实现推荐，路径可作为推荐的依据。　　偏见 Bias 问题：位置(第一更受关注)，模态(与众不同的模态吸引关注)，关键词效应(标题党的威力)　　双向选择问题：用户<->物品，稳定匹配

协同过滤算法实现：1）基于内存（Memory-based）的协同过滤（基于现有数据与简单度量运算进行推荐，可进一步细分为基于用户（User-based）与基于项目（Item-based））2）基于模型（Model-based）的协同过滤（基于现有数据训练模型，通过模型进行推荐，代表性方法如矩阵分解（Matrix Factorization）、深度学习等）

基于内存（Memory-based）：

User-based: (预测时忽略空值)：要寻找最近邻

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

Average rating of user b

计算相似度时，如果某个user对某个item未评分，则对应的r直接设为0，不用减去平均分

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

Item-base:

$$r_{ix} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{jx}}{\sum s_{ij}}$$

→ 计算方式与User-based的Similarity相似

(无需平均分修正)

相似度计算时，如果未评分，直接设为 0，不用减平均数。

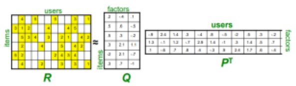
基于物品的推荐效果更好原因：物品属性单一，用户偏好多样。

基于内存优点：不受多模态非结构化信息表征与特征选取困扰；　　缺点：冷启动、稀疏性、热度偏差。

冷启动问题：(1)先非个性化推荐 (2)借助个人信息或其它网站浏览信息 (3)诱导式推荐迭代收集用户反馈 (4)与基于内容混合 (5)Side information：众包文本、知识图谱

基于模型(Model-based)：潜在因子

矩阵分解



$$\min_{P,Q} \sum_{(i,x) \in R} (r_{xi} - q_i \cdot p_x)^2$$

解决过高的 K 带来的过拟合？引入正则项，避免过大参数值。

$$\min_{P,Q} \underbrace{\sum_{training} (r_{xi} - q_i p_x)^2}_{\text{"error"}} + \underbrace{\left[\lambda_1 \sum_x \|p_x\|^2 + \lambda_2 \sum_i \|q_i\|^2 \right]}_{\text{"length"}}$$

概率矩阵分解：

- 另一种思路则着眼于参数本身的生成规律
  - 在数据稀疏且有噪声的情况下，是否可能引入某些规律，可以实现对于参数更好的描述。比如，参数符合高斯分布？
    $P(R_{ij} - U_i^T V_j | 0, \delta^2)$
  - 假设1：参数本身与噪声均符合高斯分布
  - 假设2：两个潜在因子矩阵相互独立，各用户/项目独立同分布
  - 概率矩阵分解（Probabilistic MF）

Salakhutdinov et al. Probabilistic matrix factorization. NIPS 2008

另一种思路则着眼于参数本身的生成规律

- 在概率矩阵分解中，两个参数矩阵均服从均值为0，预设方差的高斯分布
  $p(U|\sigma_U^2) = \prod_{i=1}^{N} N(U_i|0, \sigma_U^2 I)$　$p(V|\sigma_V^2) = \prod_{j=1}^{M} N(V_j|0, \sigma_V^2 I)$
- 开放问题：如何求解PMF中的矩阵参数？基于R与U、V联合分布优化求解
- 在参数完成训练后，即可简单地得到相应的完整评分矩阵
  $P(R|U, V, \sigma^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} [N(R_{ij}|U_i^T V_j, \sigma^2)]^{I_{ij}}$

非负矩阵分解：对求解过程添加非负约束，以保障元素的非负性　　矩阵分解加约束：来加入更多信息与假设，往往可以提升效果　　一种最常见的约束方式：社交约束（基于偏好相似的社交约束虽然效果很好，但未必始终成立）

情境感知的查询理解：

情境信息：帮助我们理解用户意图的信息。服务提供者可借助情境信息，为用户提供更精确的信息检索和过滤服务。

查询概念：一组有着相同语义的查询词（可以解决查询词的稀疏性问题，同时，更规范地解释查询上下文）

查询上下文可以帮助更好的理解用户查询词。　上下文感知搜索：模型准备阶段+感知查询阶段

几种常见的查询上下文类型：• 查询重组 • 查询特化 • 查询泛化 • 一般关联

情境感知的推荐任务：

推荐系统中的情境：情境是那些在重复进行同一活动时可能发生变化的因素

获取情境：显示获取（用户选）　隐式获取（从应用数据获取）　推断获取（从用户近期行为推断归纳）

情境筛选：直接询问用户　通过特征选择（主成分分析（PCA）和线性判别分析（LDA））　通过统计分析（统计测试，如信息增益、互信息、Freeman-Halton Test 检验等）　　情境感知推荐的三种范式：情境预过滤（事先根据情境信息筛选、分隔数据记录）情境后过滤 (产生推荐结果后，根据情境信息进行筛选)　情境建模　　情境信息整合到推荐系统：传统推荐：Users × Items Ratings　情境感知的推荐：Users × Items × Contexts Ratings

数据预处理：问题：1.　数据测量、采集等过程中出现的错误 2. 噪声、离群点、缺失值等数据问题 3. 重复数据

预处理方式：数据采样 维度归约 数据离散

数据采样：指选择一部分数据对象的子集进行分析的常用方法　　问题：采样缺乏代表性，将影响对于原数据集的还原程度，进而产生误导

最基本的采样技术为简单随机采用（Simple Random Sampling）

- 对于所有对象，采用简单的等概率方式进行采样
- 一般采用两种方式进行
  - 无放回采样：被采中的对象从整体中删除，仅可选中一次
  - 有放回采样：被选中的对象不会从整体中删除，可再次被选中
- 两种方式无本质区别，但有放回采样较为简单（概率不变）
- 更为复杂的方法包括分层采样（Stratified Sampling）
  - 对数据进行分组，从预先指定的组里进行采样

- 影响采样效果的重要因素之一是采样的样本容量
  - 较大的样本容量更能完整代表数据，但降低了采样的收益
  - 较小的样本容量在采样收益上更高，但可能造成信息的损失

通过分组采样的方式，可以近似确定适当的样本容量

渐进式采样从训练结果的角度确定采样规模（优点：不需要在一开始就确定采样的容量　缺点：计算开销大（需要多次迭代））

数据归约：维度归约，可以删除不具有区分度的特征，同时可能降低噪声

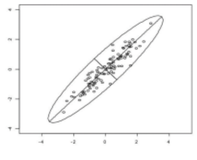
方法一：主成分分析（PCA）：通过正交变换将一组可能存在相关性的变量转换为一组线性不相关的变量，转换后的这组变量叫主成分

多维椭球：找出主轴与几个最长的轴作为新维度。

选择投影方差最大的轴。

最大特征值对应的特征向量可以最大化投影方差。

完成第一个方向基选择后，第二个方向基应选择使其不存在相关性，故优化目标为协方差为0，即 $\frac{1}{m}\sum_{i=1}^{m} a_i b_i = 0$。等价于协方差矩阵$C = \frac{1}{m} XX^T$对角化，即除对角线（方差）外的其它元素化为0。故优化目标转变为寻找矩阵P，满足PCP^T是一个对角矩阵。



最大 K 个特征值的特征向量对应的线性组合是 K 个新指标（K 个特征值的比重反映了主成分的信息量，一般>0.85）

输入：n维样本集$X = (x_1, x_2, \cdots, x_m)^T$，要降维到$n'$

输出：降维后的样本集Y

(1) 对所有样本进行中心化$x_i = x_i - \frac{1}{n}\sum_{j=1}^{n} x_j$

(2)计算样本的协方差矩阵$C = \quad XX^T$

(3)求出协方差矩阵的特征值及其对应的特征向量　归一化

(4)将特征向量按特征值大小从上到下按行排列成矩阵，取前k行组成矩阵P → 原数据/中心化后

(5)$Y = PX$即为降维到k维后的数据

PCA 特点 (1)依赖原始变量也只能反映原始变量 (2)PCA 内在假设之一是原始变量直接存在一定关联。若原始变量相互独立则降维失败，数据越相关，降维效果越好(3)PCA 的结果未必清晰可解释

方法二：　特征子集选择（而不是归纳新特征）去除冗余特征和不相关特征，或为特征赋予不同权值

数据离散 (将连续属性变换为分类属性)：(1)二元化：将连续或离散属性转化为多个二元属性 0/1　　(2)非监督离散化：不用类别信息(等宽/等频率/等深)　(3)有监督离散化（基于熵）

知识图谱概述

信息抽取含义：从语料中抽取指定的事件、事实等信息，　形成结构化的数据　　信息抽取是整合与分析的基础

信息抽取与信息检索：两者密切相关，却又存在鲜明差异 1）功能不同 • 检索：从文档集合中找文档子集 • 抽取：从文本中获取用户感兴趣的事实信息　2）处理技术不同 • 检索：可以采用统计与关键词等技术 • 抽取：需要借助自然语言处理技术　3）使用领域不同 • 检索：通常领域无关 • 抽取：通常领域相关（借助领域知识辅助抽取）

信息抽取内容：“抽取实体，确定关系”　1.实体 2.属性 3.关系 4.事件

信息抽取基本任务：命名实体 NE（最重要）、模板元素 TE（模板元素又称为实体的属性，目的在于更加清楚、完整地描述命名实体）、共指关系 CR（如果不同的命名实体表达了相同的含义，即为共指关系，也称为等价概念。共指关系的抽取任务在于抽取关于共指表达的信息）、模板关系 TR（实体之间的各种关系，又称为事实。通过关系抽取，将实体关联起来，并为推理奠定基础）、背景模板 ST（又称事件，是指实体发生的事件）

知识图谱表示：由结点和结点之间的边组成，　结点表示概念（或实体），　边表示关系（或属性）。• 在数学上，知识图谱表现为一个有向图。方向表示主被动关系，表达相互关系用双向图　　一般而言，知识图谱中的节点用来表示概念（Concept）和实体，边用来表示关系（Relation）和属性（Attribute），点和边组成知识图谱的基本单位：三元组（实体-关系-实体）　　优点：1）找到最想要的信息 2）提供最全面的摘要 3）让搜索更有深度和广度　　应用：语义搜索　问答系统　推荐系统

多模态知识图谱：　模态：　某种类型的信息或者存储信息的表示形式　　传统图谱以文本模态为主，难以描述复杂的现实世界信息

多模态知识图谱可笼统分为属性多模态与实体多模态两大类

属性多模态：实体：1）抽象概念 2）具有多模态的表示　关系：实体之间的概念联系 优点：• 易于从现有图谱中进行扩展，即通用图谱可以 轻松扩展，概念和关系不用改变 • 可以推理视觉知识 缺点：• 实体仍仅限于文本概念描述 • 1 个概念可以对应 N 张图片，但 1 个图片无法对应 N 张概念

实体多模态：实体 • 多模态的实体 关系：　1）• 逻辑关系——相同模态　2）• 物理关系——跨模态　3）• 语义关系 优点 • 视觉语义信息丰富，场景多源化 • 关系丰富 缺点 • 图谱庞大• 符号复杂

多模态知识图谱的作用：1）模态信息搜索 2）模态语义增强　　应用：推荐系统：引入多模态信息进行语义增强

知识抽取与表达

命名实体识别（NER）　基本概念：识别出文本中的人名、地名等专有名称，和有意义的时间、日期等数量短语等，并加以归类。

命名实体识别是信息抽取中的核心任务，它往往包含两个子任务：判别实体边界 + 判别实体类型　　内容：1）实体类：人名、地名、机构名　2）时间类：日期、时间　3）数值类：货币、百分比（P.S. 严格定义下哪些不属于命名实体？（部分例子）• 重复指代的普通名词：如 飞机、公司 等 • 人的团体名称以及以人命名的法律、奖项等：如 共和国、诺贝尔奖 等 • 非时间、日期、货币、百分比的数字）　　特殊的识别任务：数值识别与检测任务（百分比、钱、邮箱、时间）　时间表达识别与检测任务

难点：1）不断有新的命名实体涌现，如新的人名、地名、组织名等 2）命名实体存在严重歧义 3）命名实体构成结构复杂，如别名、缩略词、音译、包含数值的实体等 4）命名实体类型多样：如 John Smith，Mr Smith, John，实际上是共指关系

性能评价：与检索任务大致相同，采用 Precision / Recall / F-value 加以衡量　正确率与召回率的计算方式：方案 1：分子为返回的正确答案数量　方案 2：分子为返回的正确答案数量 + ½的部分正确答案数量

命名实体识别方法：1）基于词典：经常作为 NER 问题的基准算法　预先构建一个命名实体词典，出现在词典中的词汇即识别为命名实体　　词典的来源：来自于领域公开数据　　优点：简单快速，与具体语境/领域无关，容易部署和更新（只需更新词典）缺点（与基于词典/匹配分词存在类似问题）：➢ 大部分情况下很难枚举所有的命名实体名 ➢ 构建和维护词典的代价较大 ➢ 难以有效处理实体歧义　　2）基于规则：Ⅰ.采用手工构造规则模板，对符合规则的实体进行识别　　选用特征包括统计信息、标点符号、关键字、指示词和方向词、位置词　(如尾字)、中心词等　　以模式和字符串相匹配为主要手段 可借助结构/半结构化或固定表达生成模板　　该类方法的局限性明显：不同句式意味着不同的模板　　优点：当提取的规则能较精确地反映语言现象时，性能较好　　缺点：➢ 不同表达对应不同规则，导致规则库极其庞大，使用不便 ➢ 规则往往依赖于具体语言、领域和文本风格 不同领域的句法往往差异极大，如学术圈与二次元 ➢ 代价太大，系统建设周期长、移植性差而且需要建立不同领域知识库　　3）基于统计（主流）：基于统计模型的分词方法，进一步抽象而言，可以得到一个序列标注问题（四类标注：B（词的开始）、M（词的中间）、E（词的结束）、S（单字词））　　特点：1. 对特征选取的要求较高，需要从文本中选择对 NER 有影响的特征来构建特征向量　2. 通常做法是对训练语料所包含的语言信息进行统计和分析，从中挖掘出特征　3. 对语料的依赖也较大，目前缺少通用的大规模语料　4. 大部分技术仍需要进行人工标注训练数据　　Ⅰ.分支一：基于分类的命名实体识别方法词性（part-of-speech）是词汇基本的语法属性，通常也称为词类。　词性标注就是在给定句子中判定每个词的语法范畴，确定其词性并加以标注的过程。词性标注是自然语言处理中一项非常重要的基础性工作。　基本的词性标注方法：(1) 基于规则的方法：人工或通过大规模语料学习规则（核心思想是按兼类词搭配关系和上下文语境建造词类消歧规则）　(2) 基于统计模型的标注方法：HMM 等面向词序列的方法　(3) 统计方法与规则方法相结合的词性标注方法：先基于规则排除明显歧义，再基于统计模型标注，最后人工校验　　目前，更为简化的是 BIO 标签体系，其中　I 合并了 M、E 所承担的功能，即仅区分开头和后缀，不区分中间和结尾　单字实体（S）用没有任何 I 后缀的 B 取代

Ⅱ. 基于序列模型的命名实体识别方法　针对命名实体的类别不同，引入了更多、更细致的标签种类　　常用模型亦采用 HMM、CRF 以及各种序列深度学习方法（如 LSTM）等

实体对齐/实体匹配：指对于异构数据源知识库中的各个实体，找出属于现实世界中的同一实体。　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　　1.基础任务：基于表征的知识图谱实体对齐　如何使实体表征拥有统一的向量空间？联合学习结构表征和属性字符表征　TransE　结构表征：根据实体关系捕捉两个 KG 之间的实体的相似性　属性字符表征：根据属性值来捕捉实体的相似性　　2.进阶任务：多模态知识图谱的实体对齐　　如何表示不同类型的知识：通过多模态知识表征

实体链接：实体（entity），提及 or 指代（mention）　　应用：文本挖掘，知识图谱补全，信息检索和问答系统等

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

任务框架：1.提及识别：其本质与前述分词/实体识别类似 2.候选实体生成 3.候选实体排序　　方法：基于神经网络的实体链接技术 基于预训练语言模型的实体链接技术

关系的表达：1.二元组 • 适合特定领域关系抽取，类似二部图　2. 三元组 • 适合不定类型关系抽取，最为常见、基础的表达形式

基本关系抽取方法：***1.基于规则的关系抽取***（ⅰ.可以根据想抽取的关系的特点 设计针对性的规则，但部分任务可能很难制定规则 ⅱ. 需要领域专家构筑大规模的知识库，代价很大 ⅲ. 知识库构建完成后，对于特定领域的抽取具有较好的准确率，但移植到其他领域十分困难，效果往往较差）　***2. 基于模式的关系抽取***（首先由种子关系生成关系模式，然后基于关系模式抽取新的关系，得到新关 系后，从中选择可信度高的关系作为新种子，再寻找新的模式和新的关系。如此不断迭代，直到没有新的关系或新的模式产生）（优缺点：1.不同算法的差异主要在于模式生成方法和匹配方法。 2.适合某种特定的具体关系的抽取，如校长关系、首都关系。 3.基于字面的匹配，没有引入更深层次的信息，如词性、句法、语义信息等。 4.移值性差，必须为每一个具体的关系生成自己的识别模式。）　***3. 基于机器学习的关系抽取***（采用机器学习方法关系抽取模型，先通过标注语料库训练得到一个判别模型，　再利用该模型对自由文本中出现的关系实例进行识别　　往往将关系抽取问题变换为一个分类问题（二分类或者多分类），然后采用机器学习中常用的分类器来解决。　　通常采用基于特征或基于核函数的方法加以解决）（开放关系抽取 (1)基于知识监督 (2)基于句法）

基于模式的关系抽取代表方法——DIPRE：其大致思路在于先给定一些已知关系类型的种子实体对，找到出现了这 些实体对的 Occurrences，再学习 Occurrences 的模式（Pattern）。进而，根据学到的模式，寻找更多符合该模式的数据，并加入到种子集合中，不断迭代这个过程以实现关系抽取（双重迭代模式关系提取 DIPRE）　　基本元素：1) 元组：表示关系实例，如 < Foundation, Isaac Asimov>-< Title, Author> 2）模式：包含常量和变量，例如 ?x，by ?y 的形式（可表示“title” by “author”）　　基本假设：1）元组往往广泛存在于各个网页源中 2）元组的各个部分往往在位置上是接近的 3）在表示这些元组时，存在着某种重复的“模式”

DIPRE的算法流程

- 首先，输入一组种子元组实例R，如若干<title, author>的实体对
- 其次，基于种子实例集合R，找到这些元组在网页中出现的内容O（Occurrence），注意寻找的时候保留上下文信息（Surrounding Context）
- 进而，基于找到的元组实例O，生成模式P
- 最后基于生成的模式，找到更多的元组实例R
  - 此时可选择停止，或返回第二步继续基于新实例生成新模式
    - 注意，此时生成的新模式可能与之前的模式有所差异!

基于DIPRE算法，生成模式的基本步骤

- 首先，将Occurrence归纳为Order（元素的顺序）和Middle（中间部分）
- 其次，定义模式如下：
  - 模式的Order和Middle，即为 Occurrence 集合的Order和Middle
  - 模式的URLPrefix、Prefix、Suffix，分别为Occurrence集合中最长的公共（Shared）URL前缀与前、后缀。
    - 其他部分采用通配符填充

← 仅限此类网站可使用该模式

Occurrence：直译为“出现”，可以理解成元组在网页中的呈现形式 • 一般而言，只有元组的元素在网页中非常接近，才视作 Occurrence，避免因间隔太远而可能导致的语义不相关问题　　*模式*：将同一关系的不同实例在网页上所呈现的不同 Occurrence 中，相同内容保留下来，不同内容采用通配符取代，即可得到近似的模式　*URL 的前缀（Prefix）所起到的潜在作用*：模式往往仅限特定网站/体系内，跨网站则模式可能不适用　*被视作从属组织网页*：1）主机 IP 地址的前三个字段相同，如 182.61.200.X （百度） 2) URL 中的主域名段相同，如 XXX.ustc.edu.cn　*将 URL 的前缀（Prefix）引入模式中，用于描述模式的限定范围*

模式的Order和Middle，即为 Occurrence 集合的Order和Middle

模式的URLPrefix、Prefix、Suffix，分别为Occurrence集合中最长的公共URL前缀与前、后缀。

- 其他部分采用通配符填充

Snowball：仅信任支持度(满足每个模式的元组数量)和置信度高(符合该模式的元组确实符合相应关系的概率)的模式

远程监督：背景：面向文本的关系抽取方法，最大的难点在于获取足够数量的、高质量的标注　　思想：如果某个实体对之间具有某种关系，那么，所有包含这个实体对的句子都是用于描述这种关系　如果某个实体对之间具有某种关系，那么所有包含这个实体对的句子都是用于描述这种关系　　目的：获取足够数量的、高质量的标注　　语义漂移:不是所有包含该实体对的句子都表达该关系且错误不断放大　优化方案：1.动态转移矩阵：描述各个类之间相互标错的概率。(随机性较高)　2.规则学习：设计相应否定模式列表去除错误的标签（对于单一关系的判断，可以通过这种方式进行比较高效的复检。但规则 的生成成本较高） 3.注意力机制：获取对于整个句子的表示，通过此机制把最能表现关系的句子挑出来（最有效，但依赖于一个高质量的样本集合）

事件抽取：事件是信息的一种表现形式，其定义为特定的人、物，在特定时间和特定地点相互作用所产生的客观事实。　　事件包含的要素：1.事件触发词：表示事件发生的核心词，多为动词或名词（事件触发词的检测与分类是事件抽取的基本任务）2.事件类型：与触发词相对应，往往可以通过触发词分类加以识　3.事件元素：事件的参与者，主要由实体、时间等组成。 4.事件元素角色：事件元素在事件中充当的角色。　　模板：选定相应模板后，通过事件元素(事件参与者)与事件元素角色（事件元素在事件中充当的角色）的识别，将相应的元素填入模板合适的槽（描述命名实体基本信息，内容可包括名称/类别/种类）内　　***限定域事件抽取***：与预定义关系抽取存在相似之处，即预先定义好目标事件的类型及每种类型的具体结构（包含哪些具体的事件元素）　　实现方式：1.采用基于模式匹配的方法实现 2.采用基于机器学习的方法实现 3.采用远程监督等弱监督的方式，实现限定域的事件抽取　缺陷：种类有限，维护困难。　　***开放域事件抽取***：如何在开放域环境下，自动识别未知结构与类型的事件？思路一：采用无监督方法（从而摆脱对于标注语料的依赖），通过聚类找到潜在的事件簇 思路二：：与开放关系抽取中的“知识监督”方案类似

知识图谱与图计算

图学习任务：实体/关系的表征学习 关系的抽取与补全 实体/关系的对齐 基于知识图谱的推理 知识图谱应用

基于符号逻辑的知识图谱推理：***基于规则的知识图谱推理 基于符号表示的演绎推理 基于符号表示的归纳推理***

知识图谱中的类型语言：ABox TBox

基于表示学习的知识图谱推理：***基于表示学习（嵌入）的归纳推理 基于图神经网络的图谱推理 知识图谱上的关系推理任务***（链接预测 实体预测 事实三元组预测）　***知识图谱嵌入模型***（1.预测问题与推理评价 • Hit@n：所有预测样本中排名在 n 以内的比例，n 常用的取值为 1，3，10 • MR：Mean Rank 所有预测样本的平均排名 • MRR：Mean Reciprocal Rank 先对所有预测样本的排名求倒数，然后求平均　2.TransE：无法处理一对多，多对一，多对多　3.TransH：通过向量 w_r 将实体投影到关系 r 对应的超平面上，解决一对多问题　3.TransR：通过矩阵 M_r 将实体进行坐标转换到关系 r 对应的关系空间中　4. 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 DistMult RotatE ConvE）***基于图谱推理的推荐技术***

知识图谱补全：规则推理、基于表征的方法、基于路径查找的方法、引入大规模预训练模型

社会网络

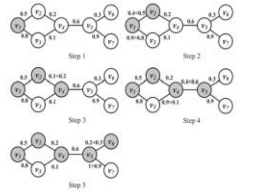
节点的角色：1.　“意见领袖”　2.“结构洞”：在于为组织引入外部的信息（衡量方式：(1) 如果一个节点，移除该节点就会使网络变成多个连通组件，则该节点即为一个结构洞　(2) 聚集系数判定：某个节点的聚集系数为：它的任意两个好友也互为好友的概率（比重）。聚集系数越低，该节点作为中介的作用也就越大　　意义：各方沟通的桥梁，结构洞具有排他性)

基本传播模型：1.信息级联发生在有向图上（对于无向图，可以将其连边转化为双向边进行处理）2.　每个节点仅能将信息传递给与其直接相连的节点 3. 网络中节点的状态是二元（Binary）的（激活/未激活）(关于激活：(1) 接收到信息，并且尝试将信息传给别人才叫激活　(2) 激活能力有一定的时限　(3) 已激活的节点才具备激活其他节点的能力　(4) 激活是不可逆的过程)

独立级联模型：每次激活都是一次独立事件，相互不产生影响，同时，每个已激活节点，只有一次机会尝试激活他/她的未激活邻居节点　　对于节点 v 而言，他激活邻居节点 w 的概率采用 Pvw 表示（基本传播模型里，为简化考虑，一般将 Pvw 设为 1/N，N 为 w 节点的入边的数量，此外还有特殊情况，如带权图）　　整个传播过程直到所有节点都被激活，或没有新节点可以被激活为止

Algorithm 7.1 Independent Cascade Model (ICM)

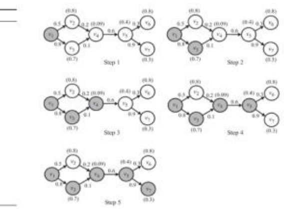
```
Require: Diffusion graph G(V,E), set of initial activated nodes A_0, activation probabilities p_{v,w}
return Final set of activated nodes A_∞
i = 0;
while A_i ≠ [] do

  i = i + 1;
  A_i = [];
  for all v ∈ A_{i-1} do
    for all w neighbor of v, w ∉ ∪_{j=0}^{i} A_j do
      rand = generate a random number in [0,1];
      if rand < p_{v,w} then
        activate w;
        A_i = A_i ∪ {w};
      end if
    end for
  end for
end while
A_∞ = ∪_{j=0}^{i} A_j;
Return A_∞;
```



线性阈值模型：一个用户会被某个信息激活，如果来自他已激活邻居的影响超过某个阈值　　阈值预先设定，往往为从[0,1]均匀分布中随机抽取的一个数值

Algorithm 8.1 Linear Threshold Model (LTM)

```
Require: Graph G(V,E), set of initial activated nodes A_0
return Final set of activated nodes A_∞
i=0;
Uniformly assign random thresholds θ_v from the interval [0, 1];
while i = 0 or (A_{i-1} ≠ A_i, i ≥ 1) do
  A_{i+1} = A_i
  inactive = V − A_i;
  for all v ∈ inactive do
    if Σ_{j connected to v, j∈A_i} w_{j,v} ≥ θ_v then
      activate v;
      A_{i+1} = A_{i+1} ∪ {v};
    end if
  end for
  i = i + 1;
end while
A_∞ = A_i;
Return A_∞;
```



线性阈值模型与独立级联模型的区别：随机性 1）对于独立级联模型来说，其随机性在于抛硬币的过程　2）对于线性阈值模型来说，其随机性在于边权重/阈值的确定

传播最大化问题：假定初始的种子节点集合为 S，预期激活的节点集合为 f(S)，信息传播最大化的目的，在于限定 S 集合的前提下，最大化 f(S) 的规模（常见的约束为限定 S 集合的规模，即 |S| 。如果 S 集合中的节点价值不等，则可将约束进一步扩展）　　解决传播最大化问题的启发式思路：寻找网络中最具影响力的节点　　启发式方法：1. PageRank 及其衍生模型（如 HITS） 2.核心性度量（如度（Degree）、紧密度（Closeness）、介数（Betweenness）) 3.计算单个节点所能够激活的邻居数量，再进行排序　　子模特性：1. f(S) 函数是非负的　2. f(S) 函数是单调非减的，即 f(S+v)>=f(S)　3. f(S) 函数是具有子模特性（Submodularity）的，即：对于任何集合对 S，T，且满足 S ⊆ T 时，给定节点 v，有 f(S∪{v})-f(S)>=f(T∪{v})-f(T)　　传播最大化问题是 NP-hard

基于贪心算法的传播最大化问题求解伪代码：

Algorithm 7.2 Maximizing the spread of cascades – Greedy algorithm

```
Require: Diffusion graph G(V,E), budget k
return Seed set S (set of initially activated nodes)
i = 0;
S = [];
while i ≠ k do
  v = arg max_{v∈V\S} f(S ∪ {v});
  or equivalently arg max_{v∈V\S} f(S ∪ {v}) − f(s)
  S = S ∪ {v};
  i = i + 1;
end while
Return S;
```

衍生传播问题：独立级联模型的松弛版本：稳定状态传播　节点状态不再二元化，而是引入一个变量表示当前被激活的概率　如果被激活概率不为 0，则节点可以持续对外输出信息 / 影响　　出现问题：(1) 不合理的反向传播：解决办法：1.遍历整个网络（费时费力）2.启发式方法，限制每个节点的迭代次数，超过一定轮次阈值，则该节点被激活的概率 不再更新（阈值大致根据节点到信息源的最长 / 短路径决定）　(2)信息传播与接收的捆绑：解决办法：1.修改目标函数（核心假设：如果某个节点被激活，那么他的所有邻居都被覆盖　贪心算法求解）2.修改模型框架：单独对信息接受过程进行建模

网络爬虫

反爬虫策略：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。应对策略：利用 Python 的 Request 库允许用户自定义请求头信息的手段。在请求头信息中将 User-Agent 的值改为浏览器的请求头标识，从而绕开反爬虫机制

```
import requests
# 伪造请求头信息，欺骗服务器
headers = {"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:9527.0) Gecko/20100101 Firefox/9527.0"}
resp = requests.get("http://127.0.0.1", headers=headers)
print(resp.status_code)
```

2. IP/账号访问次数/频率：通过限制特定 IP 地址/账号访问频率和次数进行反爬。其本质在于判断浏览行为是否是人类行为。应对手段：1)构造 IP 代理池，然后每次访问时随机选择代理（有相关服务，通过各种提供免费 IP 的网站来提供代理池） 2)每次爬取行为后间隔一段时间 3)注册多个账号以保障数据收集　挑战：账号本身的成本问题、账号被查封的危险

3. 验证码：通过各类验证码，判断浏览者属于人类还是机器(从简单的字符识别到 复杂的逻辑推理) 应对手段：1)简单的字符识别：基于机器学习与模式识别相关技术 2)复杂的逻辑推理：人工辅助破解（focus：验证码是一把双刃剑，在抵御爬虫的同时也在伤害用户体验）

4. 动态网页：从网页的 url 加载网页的源代码之后，会在浏览器里执行 JavaScript 程序（网页内容由脚本加载，而直接抓取则只能得到空白页面； 此类情况在抓去在线播放的音视频文件时尤为常见）应对手段：核心思路：模拟调用请求 • 使用审查元素分析 ajax 请求，如此循环直到获得包含数据信息的 json 文件

5. 蜜罐技术 • 网页上会故意留下一些人类看不到或者绝对不会点击的链接。由于爬虫会从源代码中获取内容，所以爬虫可能会访问这样的链接。只要网站发现有 IP 访问这个链接，立刻永久封禁访问者，从而难以继续爬取 应对手段：核心思路：干涉爬虫路径 • 通过工具库判断页面上的隐含元素，使爬虫避开这些元素，可以部分回避蜜罐的诱导。

6. 用户权限限制：不同类型/级别的用户给予不同的内容权限 • VIP、SVIP、蓝钻、红钻、绿钻、各种钻……• 应对手段： • 氪金，就可以变强

7. 其他的反爬虫策略 • 不同的网页结构：每个相同类型的页面的源代码格式均不相同 • 同样也是双刃剑：增加爬取难度的同时降低用户浏览体验（逼死强迫症） • 多模态的呈现方式：文字转为图像或视频 • 应对策略：OCR、语音识别、图像/视频标签技术

网页文字处理

一.基于字符匹配的方法（机械分词方法）

正向最大匹配分词（FMM）反向最大匹配分词（RMM）双向最大匹配分词（BM：FMM+RMM）最少切分分词（最短路径分词）　ASM(d,a,m)　d，表示匹配方向，+1 为正向，-1 为逆向 • a：每次匹配失败后增/减字符数，+1 为增字，-1 为减字 • m：最大/最小匹配表示，+1 为最大匹配，-1 为最小匹配 • 例如，ASM（+，-，+）即正向减字最大匹配（即 FMM 方法） 现代汉语最大匹配更实用

FMM：从左至右尽可能查找最长的词　分词速度较快，但错误率较高

RMM：从右至左尽可能查找最长的词 统计证实 RMM 分词效果更好 （由于汉语中偏正结构较多， 若从后向前匹配，可以适当提高精确度）

BM：综合比较 FMM 与 RMM 两种方法的切分效果，从而选择正确的切分 有助于识别分词中的交叉歧义

最少切分分词方法：使句子中切出的词数目最少。 • 等价于在有向图中搜索最短路径的问题。 • 将每个字元视作节点，每个词形成一条边。 • 边权重可都视为 1，也可根据词频决定（尽量切出高频词） • 结合权重/概率之后，实际上可视作基于统计的分词方法　N-最短路径法：保留 N 条最短的路径，以提供更多分词方案（求一个图中两点的最短 N 个路径：Dijkstra（O(n^2)））

基于匹配分词方法的优劣 • 优点：效率高、直观性好 • 缺点：对词典的依赖性 • 维护高质量词典需要极大的开支 • 永远难以应对新生词汇 • 词汇频率/重要性往往对结果不产生影响

二.基于统计的分词方法

$$P(\boldsymbol{w}|\boldsymbol{c}) = \max\{P(\boldsymbol{w}_a|\boldsymbol{c}), P(\boldsymbol{w}_b|\boldsymbol{c}), \dots, P(\boldsymbol{w}_k|\boldsymbol{c})\}$$

一般化过程：1. 建立统计语言模型 2. 对句子按不同方案进行分词 3. 计算不同分词方案的概率，选出概率最大的分词结果 • 理论上，基于统计的分词方法可以不需要词典，但实际应用中第 2 步可以采 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、自动消除歧义的优点。

N-gram 模型

N = 1，一元文法模型（最大概率模型），$P(w) = P(w_1)P(w_2)\dots P(w_n)$

N = 2，Bigram模型，$P(w) = P(w_1)P(w_2|w_1)\dots P(w_n|w_{n-1})$

以Bigram模型为例，基于最大似然估计进行推断

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

• 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。

以Bigram模型为例

1. 首先，构造训练语料库，计算所有的$C(w_n)$与$C(w_{n-1}w_n)$。
2. 其次，对于每一个可能的分词序列w，计算以下公式
   • $P(w) = P(w_1)P(w_2|w_1)\dots P(w_n|w_{n-1})$
   • 其中，$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$
3. 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。

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

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劣：• 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

三.　隐马尔可夫模型（HMM）

• 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

停用词： Stopwords，指文档中频繁出现或对实际语义影响不大的词语。去停用词原因：• 重复率很高，会造成索引中的倒排表很长，影响查询性能。 • 对最后结果的排序没什么贡献，反而可能产生干扰。

常用的停用词识别方法 • 较为成熟的停用词识别方法有：文本频率、词频统计、熵计算等。 • 更为复杂的算法将结合统计与句法或内容分析。

现代搜索引擎的趋势是逐渐减少对停用词的使用。 • 现代搜索引擎更关注利用语言的统计特性来处理常见词问题。 • 采用压缩技术，降低停用词表的存储开支。 • 引入词项权重，将高频词的影响降至最低。 • 索引去除技术，低于权重的词项将被排除 ο

归一化处理：归一化/词根化，指还原词语的特殊形式的过程。可以有效降低词项的数量并减少歧义

词干提取：Stemming，指去除单词前后缀，获得词根的过程。常见的前后词缀有“复数形式”、“过去分词”、“进行时”等

词形还原：Lemmatisation，指基于词典，将单词的复杂形态转变成最基础的形态。并不是简单地将前后缀去掉，而是会根据词典将单词进行转换

词干提取与词形还原的相同点：1)目标一致。两者的目标均为将词的不同形态简化或归并为基础形式。2)结果交叠。两者不是互斥关系，其结果有部分交叉。3)方法类似。目前两者主流方法均是利用语言中的规则或词典实现。

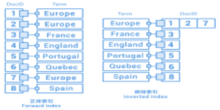
•词干提取与词形还原的不同点 1)在原理上，词干提取采用“缩减”，而词形还原采用“转变”。 2)在复杂性上，词形还原需考虑词缀转化、词性识别等，更为复杂。 3)在实现上，词干提取主要利用规则变化，而词形还原更依赖于词典。 4)在结果上，词干提取不一定得到完整单词，而词形还原是完整单词。

基本词干提取方法：Porter Stemming 基本流程：1. 去除单词的复数形式 2. 去除 -ed (ly) 或 -ing (ly) 等后缀 3. 将 –y 改为 – i 4. 处理双重后缀，如 –ization 等 5. 处理 –full ， – ness 等后缀 6. 处理 –ant ， – ence 等后缀 7. 去除掉最后的 – e 和 – ll

网页索引

倒排表（Posting List）：文档 ID 列表，列举词项在哪些文档中出现

	正排	倒排
关键索引（key）	Document	Term
优点	易维护	构建、维护成本高
缺点	搜索耗时长	搜索快



倒排表三步走：1.检索每篇文档，获得对，并写入临时索引 2. 对临时索引中的词项进行排序 3. 遍历临时索引，对于相同词项的文档 ID 进行合并　基于倒排表的查询：如果两个倒排表的长度分别为 x 和 y，则合并供需 O(x+y)次操作　同时扫描两个倒排表，所需时间与倒排记录的数量呈线性关系

优化问题：1. AND 是合并操作，先处理文档频率小的，再处理大的 2.任意组合的布尔查询：首先，获得所有词项的文档频率 • 其次，保守地估计出每个 OR 操作后的结果大小 • 考虑 x+y 的最坏情况 • 最后，按照结果从小到大的顺序执行 AND 3.跳表指针：1)如何设置 Ⅰ.较多的指针+较短的步长→更多的跳跃机会（耗费更多的存储空间） Ⅱ. 较少的指针+较长的步长→更少的指针比较次数（存储空间消耗更少，但跳跃机会也更少） 2)如果倒排表长度为 L，则间隔 L^(½)均匀放置跳表指针 3) 索引的动态变化也会影响跳表指针的设置 • 如果索引相对固定，建立有效的跳表指针就相对容易 • 反之，经常更新的索引很难建立合适的跳表指针

扩展性问题：

1）除了文档 ID，搜索引擎往往在倒排表中加入更多元素（词项频率（Term Frequency）、词项类型等） 2）除了基本的词查询之外，倒排表还可能面临更多需求（词组查询、词项邻近、文档区域）　短语查询的需求解决方案：1) 二元词索引 (问题：词汇表迅速增长)：将文档中每个连续词对看成一个短语，构建面向二元词的倒排表，并处理两个词（或多个词）构成的短语查询 更长的短语查询：可以分成多个短查询来处理（如果采用二元词索引拼接的方式，对于该布尔查询返回的文档，不能确定其中是否真正包含最原始的四词短语）　2）位置信息索引：在记录词项的同时，记录它们在文档中出现的位置，可以达成更广泛的查询。对于短语查询，仍采用合并算法（AND），查找符合的文档。当然不只是简单地判断两个词是否出现在同一文档中，还需要检查他们出现的位置情况是否符合要求。并且该方法能够用于邻近搜索（例如，间隔 k 个词）

索引存储：词典与倒排表分开存储（可以支持并行、分布式查询）　词汇表存储：顺序、哈希、B/B+树、Trie 树（前缀树）

索引压缩：

（节省磁盘空间（￥），提高效率（内存利用率或数据传输速度） 前提：快速的解压缩算法。目前的启发式算法效率都比较高。）意义：1）对词典：压缩得足够小，可以直接放入内存中，提升效率 2）对倒排记录表：减少所需的额磁盘空间，可以更多移入内存，并且可以减少从磁盘读取倒排表所需的时间

词典基本存储方式：定长存储（词项本身设置定长为 20 字节，另外需要记录文档频率和指向倒排表各 4 字节），但是空间会浪费（英文中平均的词典词项长度为 8 个字符）

压缩词项列表：将词典视作单一字符串（在词项字符串中，每个词项平均长度 8 个字节 • 其次，词项文档频率与倒排表指针各 4 字节不变 • 最后，词项指针约 3 字节）

按块存储：单一字符串在词项指针上需要占用较多额外空间。通过为每 k 个词项存储一个指针，来减少指针的总数量。需要额外 1 个字节用于表示词项长度。例如，当块的大小 k=4 时 • 不采用按块存储时，每个指针花费 3 字节，共需 12 字节 • 采用按块存储时，只需要花费 3+4*1=7 字节　在搜索上的问题：二分查找只能在块外进行，块内采用线性查找方式（随着 k 上升，线性查找部分增多，效率更低）　改进：前端编码（Front Coding）

8automata8automate9automatic10automation

→8automat*a1◊e2◊ic3◊ion

编码 automat　除automat外的额外长度

采用间距代替文档 ID——再进一步：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

查询和评估

• 两个视角下对于数据集的切分
  • P/N：Positive or Negative，表示算法对样本的判断
  • T/F：True or False，表示算法判断的正确与否（和结果是否一致）
  • 四种简写的含义：
    • TP：True Positive，样本为正例，且被判定为正，即真正
    • FN：False Negative，样本为正例，但错误地被判定为负，即假负
    • FP：False Positive，样本为负例，但错误地被判定为正，即假正
    • TN：True Negative，样本为负例，且被判定为负，即真负

	被检索文档	未检索文档
相关文档	TP	FN
不相关文档	FP	TN

面向单查询的基本评价指标： 1）准确率（Precision）：指检索出的文档中，相关文档所占的比例，也称查准率。计算公式为 TP/(TP+FP)　2）召回率（Recall）：指所有相关文档中，被检索出来的部分的比例，也称查全率。计算公式为 TP/(TP+FN)　召回率的近似计算：缓冲池（Pooling）针对某一检索问题，各个算法分别给出检索结果中的 Top N 个文档，将这些结果汇集起来并进行人工标注，从而得到一个相关的文档池（虽然它实际上仍然无法得到全部相关文档，因此并不能得到召回率的绝对值。但是，它可以比较各个算法的相对优。N 通常取 50-200）

F 值：

F值（F-measure），即准确率与召回率的加权调和平均数

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

通常情况下，我们取α=0.5或β=1（即两者同等重要）

• 此时，可得基本的F1值，即F=2PR/(P+R)

P-R 曲线：可以直观地看出准确率 与召回率之间的平衡关系

ROC 曲线（Receiver Operating Characteristic Curve，接受者操作特征曲线）以真正率【TP/(TP+FN)】和假正率【FP/(FP+TN)】作为两条轴线。对角线表示区分能力为 0，即随机猜测。在对角线上端越远，效果越好。低于对角线的结果无意义（无区分度）

如何基于P-R曲线或ROC曲线判别算法好坏
• 如果线A将线B完全包住，显然线A对应的算法效果更好
• 如果两条线发生重合，则可依据以下规则判别：
  • 计算AUC，AUC更高者效果更好
    • Area under curve，即曲线下面积
    • 可通过积分近似计算
  • 另外，当使用P-R曲线时，可使用平衡点计算
    • 平衡点即Precision = Recall的点，值越高越好

P-R曲线与ROC曲线的选择
• ROC曲线兼顾正负样例，更为全面，而P-R曲线则只考虑正例
  • 用户往往更关心正样本，如果面向特定应用场景（如检索），P-R曲线是个好选择
• 单份数据正负样本比例失调时，P-R曲线更合适
  • 当负样本比重过高时，负例的数目众多致使FPR的增长不明显，导致ROC曲线呈现一个过分乐观的效果估计，从而难以体现出性能的差异性

P-R曲线与ROC曲线的选择
• P-R曲线受分布影响大，多份数据且正负比例不一时ROC曲线更合适
  • ROC曲线两个指标各自针对正负样本，而Precision只针对正样本，受影响较大

P@N，即 Precision@N，指前 N 个检索结果文档的准确率：如果相关文档数小于 N，P@N 的理论上限必定小于 1

R@N，即 Recall@N，指前 N 个检索结果找回的相关文档比例 ：1）由于返回结果有限，Recall@N 值，甚至其理论上限往往都远小于 1　2）理论上限为 N/相关文档数，即使通过 Pooling 加以控制仍然较小

R-Precision=P@相关文档总数

未插值AP：查询Q共有6个相关结果，排序返回了5篇相关文档，位置分别是第1，第2，第5，第10，第20位，则AP=(1/1+2/2+3/5+4/10+5/20+0)/6（注意补0）

插值AP：事先选定插值点数并进行插值。例如，当我们计算11点平均时，计算在召回率分别为0（第一条），10%，20%，…，100%的十一个点上的正确率求平均

简化AP：只对返回的相关文档进行计算，AP=(1/1+2/2+3/5+4/10+5/20)/5，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）

累计增益 CG：位置 1 到 p 的检索结果相关度之和;　折损累计增益 DCG:若把相关度高的文档排在后面给惩罚;

$$DCG_p = rel_1 + \sum_{i=2}^{p}\frac{rel_i}{log_2(i)}$$ （ppt 用第一个）　或 $$DCG_p = \sum_{i=1}^{p}\frac{2^{rel_i}-1}{log_2(i+1)}$$

假设有10个文档，相关度为0-3之间，10 个文档的得分依次如下：
• 3, 2, 3, 0, 0, 1, 2, 2, 3, 0
理想的输出结果序列为：3, 3, 3, 2, 2, 2, 1, 0, 0, 0
• 由此计算 iDCG依次为：3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10.88
而与此同时，基本的DCG结果如下：
• 3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61（单调非减特性）
由此可得 NDCG结果如下：1, 0.83, 0.87, 0.76, 0.71, 0.69, 0.73, 0.8, 0.88, 0.88
• 可以看到任何查询结果位置的NDCG 值都规范化为[0,1]之间的值

其中 iDCG 是根据文档相关性从大到小排序得到理想化的最序列，并对此序列计算 DCG 值所得到的。

NDCG = DCG / iDCG

多查询指标： 1）MAP：对所有查询的（简易）AP 求算数平均（可以反映全部查询的综合效果，但在查询难度不平衡的条件下有误导） 2）MRR：多个查询所得的倒数(第一相关文档)排序求评价（eg. 两个查询,第一个的第一个相关文档在位置 2,第二个的第一个相关查询在位置 4,则 MRR=(1/2+1/4)/2=3/8，即平均在 8/3 的位置上找到第一个相关文档） 3）方差：一个检索系统对不同查询的方差通常大于多个检索系统对相同查询的方差，说明查询难度差异大

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)作为一个整体，应满足以下条件： • R(q)中所有的结果文档都与查询 q 本身有较大的相关性。 • 总体上要有较小的冗余度，以覆盖 q 的不同方面。）　核心思想：降低用户无法获得所需信息的风险（尽可能确保排序靠前的结果中至少有一个结果满足用户的需求）　总体需求：衡量不同文档之间的主题差异性。两种衡量方式： 1）隐式模型：只计算文档之间的差异性 • 文档是什么内容，不会也无法进行详细考量 2）显式模型：更加具体地考量文档所对应的用户意图 • 会从文档中抽取主题，并显式地实现主题的多样化

• 隐式模型代表：MMR
• 最大边界相关性（Maximal Marginal Relevance，MMR）
  • 最早的相关性衡量模型，由Carbonell和Goldstein在1998年提出
    $MMR^{def} = Argmax_{d_i \in R \setminus S}[\lambda P(d_i|q) - (1-\lambda)\max_{d_j \in S} P(d_i|d_j)]$
  • 该公式由两部分组成，其中：
    • 前一半表示文档集与查询 q 的相似性
    • 后一半表示文档之间的多样性
    • λ 用于调节两部分之间的比例
    • 文档可以采用tf-idf，或者word2vec等文本表征工具进行表征

• 显式模型代表：FM-LDA
• Facet model with LDA (FM-LDA)
  • 由Carterette与Chandar在2009年提出，考虑文档的不同子主题（Facet）
    $L(y_i|F,D) = \prod_{j=1}^{m}\left(1 - \prod_{i=1}^{n}(1 - p(f_j \in d_i))^{y_i}\right)$
  • 其中，$p(f_j \in d_i)$表示文档Di包含主题Fj的概率，Yi=1表示某文档被选中
  • 由此，联乘部分表示主题Fj至少被一个文档所涵盖的概率
  • 当有约束$\sum y_i \le l$时，返回的文档数量被限定为 l 篇

相关性反馈：用户在查询后标记相关/不相关，然后迭代更新查询，以得更好的结果　过程：1. 首先，用户提出一个查询条件（Query） 2. 对于返回的文档，用户标出相关与不相关的部分 3. 系统根据用户反馈，获得用户信息需求更为准确的描述 (a) 基于相关性反馈，更新查询条件 b) 基于新查询条件，获取新的结果文档并再次提交用户进行评估）➢ 上述过程将根据情况进行一次或多次的迭代，从而不断接近最优查询条　目的：通过相关性反馈，获得用于表达用户查询意图的最优查询条件（常见方式：为已有词项 添加不同权重，或增加 新的词项 • 这一过程应对用户隐藏）　相关性反馈类型： 1）显式反馈：用户显式地参与交互过程（如用户点击记录，但是只有正样本，并且用户不点击，不代表完全不相关。所以有了拓展的显式反馈：收集负面评价的渠道日益丰富，还有用户评论） 2）隐式反馈（更常见）：系统追踪用户行为来推测返回文档的相关性（判定不一定很准确，但省却了用户的显式参与行为。如鼠标键盘动作，用户眼球动作等）优点： • 不需要用户显式参与，减轻用户负担，提升用户体验 • 用户行为某种程度上可以反应其兴趣，因此具有可行性 缺点： • 对行为分析有着较高的要求 • 准确度难以保证 • 某些情况下需要增加额外设备（且很贵！） 3）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 （准确性差，可能查询漂移） 4）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为完整的查询条件。 • 暗含的功能：用户选择和确认的查询扩展能够更好表达其查询意图。　实现：利用同义词辞典，可以实现查询条件的扩展。一般而言，查询扩展有助于提升查询的召回率（找得更全）。但是，可能会影响准确率，尤其在扩展词存在歧义的情况下。编纂和维护同义词词典需要很大的代价 。　类型：1.利用人工编纂的同义词词典 2.全局分析与同义词词典的自动生成 3. 基于搜索日志进行优化　自动构建相关词词典的两种思路（通过分析文档集中的词项分布，来自动生成同义词/近义词词典）：1.如果两个词经常和相似的词共同出现，则它们很可能是相似的 2.如果两个词经常与相同的词以一种特定的语义关系共同出现，那 么他们很可能是相似的　基于搜索日志的查询扩展：搜索日志目前是搜索引擎查询扩展的重要方式（潜在扩展查询 1.查了一个词又查另一个（后是前的） 2.查了两个词（互为）后都点进同一个链接）　相关词扩展的潜在问题：词项关联的质量是一个问题（1. 有歧义的查询词可能导致统计上相关，而意思上不相关的词 2. 同时，由于扩展的查询词与原查询词高度相关，扩展后的查询也未必能够 获得更多的相关文档）

排序

排序的难点：Web 网页的信息组织与内容质量参差不齐，用户庞杂且缺乏知识和经验，用户意图多样差异巨大

好的排序：网页内容匹配程度+网页内容质量

信息检索模型：文档与查询的表示形式与相关性的框架，实质是对文档基于相关性进行排序，在理解用户的基础之上产生近似用户决策的结果从而在顶部返回最相关的信息。[D,Q,F,R(Di,q)]　布尔模型 D:文档表达-词项的组合；Q:查询表达-布尔表达式；F:完全(二值)匹配；R:满足布尔表达式,相关性为 1(否则 0)

相关度计算：Jaccard 系数：$$JACCARD(A,B) = \frac{|A \cap B|}{|A \cup B|}$$　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：$idf_t = log_{10}\frac{N}{df_t}$

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

向量空间模型 VSM：文档和查询视作一个词项权重构成的向量->计算向量之间相似性->按相似度大小排序返回 Top-K 文档给用户。(D:文档表达，每个文档可视作一个向量，其中每一维对应词项的 tf-idf 值；Q:查询表达，可视作一个向量，其中每一维对应词项的 tf-idf 值；F:非完全匹配方式；R:使用两个向量之间的相似度来度量文档与查询之间的相关性）　优点：简洁直观，可支持多种不同度量和权重方式，实用效果不错；　缺点：缺乏语义层面的理解和匹配，同时依赖 tf-idf 值也可能造成干扰，用户无法描述词项之间的关系，词项之间的独立性假设实际上不成立。

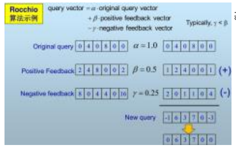
文档匹配与迭代优化：

欧氏距离（不行，太逊了。对于向量长度（文档长度）非常敏感） $dist(\vec{x},\vec{y}) = \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2 + \dots + (x_n-y_n)^2}$

余弦相似度（文档向量与查询向量的夹角大小来计算）：$\cos(\vec{q},\vec{d}) = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$

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，Dr 为已知相关文档的向量集合，Dnr 为已知不相关文档的向量集合 • q0 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。　相关反馈信息融入到向量空间模型



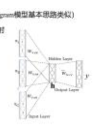
基于迭代的查询意图更新 ： 正反馈 vs 负反馈

• 正反馈的价值往往大于负反馈 用户更关心符合需求的标准答案，而不是错误答案。相应的，可以通过设置 β > γ 来给予正反馈更大的权重

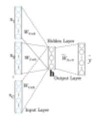
• 很多系统甚至只允许正反馈，即 γ = 0 • 收集真正的负反馈往往比较困难

Word2vec：1. 根据上下文预测中心词（CBOW） 2. 根据中心词预测上下文（Skip-gram）

• 我们以CBOW模型为例，介绍其工作原理（Skip-gram模型基本思路类似）
• CBOW本质上是获得上下文的词项→中心词的映射
  • 如何获得这样的映射？
  • 基于深度学习的CBOW算法详细步骤：
    • 使用one-hot（0/1）向量表示词项：x_i
    • 计算隐藏层（深度学习基操）：
      $h = \frac{1}{C}W^T(x_1 + x_2 + \dots + x_C) = \frac{1}{C}(v_{w_1} + v_{w_2} + \dots + v_{w_C})^T$



• CBOW算法详细步骤：
  • 计算预测的中心词的概率：
    $\hat{y} = softmax(h\hat{W})$
  • 计算交叉熵损失，反向传播优化模型：
    $L = -\sum y_j \log(\hat{y}_j)$
  • 基于多个不同滑动窗口进行多次训练
  • 最终得到各个词项的表征：Wx_i



两个模型比较：1）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　2）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 优缺点：1）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　2）缺点 • 无法解决一次多义的问题，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

排序学习算法： Pointwise 算法　Pairwise 算法　Listwise 算法

Pointwise： 基本假设：训练样本中的任何一个查询-文档对，都可以映射到一个分值或一个有序的类别（如优良中差） • 相应的，给定一个查询-文档对，Pointwise LTR 将试图预测其分值/类别。 • 常见的模型类别包括： 1）回归，将查询-文档对映射到具体分数 2）分类，将排序问题转化为一个面向有序类别的二分类/多分类问题　3）有序回归，在映射到具体分数的同时保持样本之间的有序关系　可以简单且广泛地套用已有回归、分类算法　局限性：1）首先，Pointwise 类方法往往更为注重文档的相关度得分，而并不注重文档之间的相关性排序（Pairwise 类方法的出现，为解决这一问题提供了新的手段）2）其次，不同查询所对应的文档，尤其相关文档数量不同，对损失函数的贡献也各不相同，一定程度上影响效果

Pairwise： 基本假设：同样是将排序问题转化为分类问题（二分类或三分类） • 每次比较一个查询与两个文档，衡量两个文档的偏序（Partial Order） • 分类器的目的在于判断哪个文档应该排在前面（对应的标签为{1, -1}或{1, 0, -1}（0 表示两个文档可以并列））　相比于 Pointwise 类算法，Pairwise 类排序算法通过衡量样本之间的偏序关系，实现了从绝对相关性（分值）到相对偏序的改进　缺陷：1）两两成对导致样本数大为提升，计算资源开支增加　2）Pairwise 类算法仍然受样本不平衡问题的影响　3）最后，Pairwise 类算法无法体现全局排序的合理性。综上，由此引出了最后一类算法：Listwise 类算法　Pairwise 局限性：样本不平衡性的影响以及算法无法体现全局排序

Listwise：基本思路：直接面向整体排序结果进行优化（直接将排序的完整队列作为学习的对象） 解决这一问题采用以下两种思路：1）直接采用某种 IR 指标对排序进行优化 2）直接设计面向完整排序的损失函数　代表性思路：采用某种 IR 指标对排序进行优化　通常情况下，由于全局优化的作用，Listwise 类排序算法可以取得相比于 Pointwise 和 Pairwise 类算法更好的效果。 小的挑战：例如两个网页并列的情况（相当于 Pairwise 中文档对标签为 0 的情况）　拓展：Listwise 与 Pairwise 的融合：FocusedRank

PageRank：优质网页引用或推荐的网页一定也优质

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

PR(pi)为网页 pi 的 PageRank 值，初值为 1/N；PR(pi)为指向网页 pi 的某个网页 pj 的 PageRank 值；L(pj)为网页 pj 发出的链接数量；d 为阻尼系数，取值在 0-1 之间；N 为网页总数，M(pi)为链入 pi 的页面集合。

邻接矩阵->跳转矩阵：转置（行列互换）后各数值除以出边数量

$$A = \begin{bmatrix} 0,1,1,1,1,0,1; \\ 1,0,0,0,0,0,0; \\ 1,1,0,0,0,0,0; \\ 0,1,1,0,1,0,0; \\ 1,0,1,1,0,1,0; \\ 1,0,0,0,1,0,0; \\ 0,0,0,0,1,0,0; \end{bmatrix} \Rightarrow R = \begin{bmatrix} 0, & 1, & 1/2, & 0, & 1/4, & 1/2, & 0; \\ 1/5, & 0, & 1/2, & 1/3, & 0, & 0, & 0; \\ 1/5, & 0, & 0, & 1/3, & 1/4, & 0, & 0; \\ 1/5, & 0, & 0, & 0, & 1/4, & 0, & 0; \\ 1/5, & 0, & 0, & 1/3, & 0, & 1/2, & 1; \\ 0, & 0, & 0, & 0, & 1/4, & 0, & 0; \\ 1/5, & 0, & 0, & 0, & 0, & 0, & 0; \end{bmatrix} \quad M = \begin{bmatrix} 1/7 \\ 1/7 \\ 1/7 \\ 1/7 \\ 1/7 \\ 1/7 \\ 1/7 \end{bmatrix}$$

$$R \cdot M = \begin{bmatrix} 0.321 \\ 0.148 \\ 0.148 \\ 0.064 \\ 0.148 \\ 0.036 \\ 0.029 \end{bmatrix} = M'$$

若 M 与 M’ 对应维元素差值高于阈值则继续迭代至收敛

陷阱节点：只有一条指向自己的边，没有其它出边；　终止节点:没有出边，如同黑洞；　孤立结点：没有任何入边，单纯使用 markov 链无法跳转，仅有初始概率，不再更新也不影响其它结点。　解决：重启动：d 或 alpha （随机跳转概率是 1-d）。why? 其中的(1-d)/N 的部分相当于以一定等概率被选中作为新起点，由此跳出了陷阱和黑洞的干扰（所有节点全联通）。　收敛?马尔可夫过程的三个收敛条件（转移矩阵 A 所有元素非负，且每一列元素和为 1，为马尔科夫矩阵；Restart 保障图是强连通，则 A 不可约；A 非周期）Pagerank 算法都满足

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　解特征方程（A-I）x=0 且$\sum x_i = 1$ 得概率 x。(λ=1)

则有$P_{n+1} = AP_n$，形成马尔科夫过程