1. 简述搜索引擎工作原理

答:

1）收集因特网上几千万到几十亿个网页并对网页中的每一个词(即关键字)进行索引，建立索引数据库的全文搜索引擎。

2）当用户查找某个关键词的时候，所有在页面内容中包含了该关键词的网页都将作为搜索结果被搜出来。

3）在经过复杂的算法进行排序后，这些结果将按照与搜索关键词的相关度高低，依次排列

1. **爬行和抓取**
   1. 爬虫程序（Spider，robot ）
   2. 搜索引擎从已知的数据库出发，就像正常用户的[浏览器](http://baike.baidu.com/view/7718.htm)一样访问这些网页并抓取文件。
   3. 搜索引擎通过这些爬虫去爬互联网上的外链，从这个网站爬到另一个网站，去跟踪网页中的链接，访问更多的网页
   4. 这些新的网址会被存入数据库等待搜索。
2. 建立索引
   1. 蜘蛛抓取的页面文件分解、分析，并以巨大表格的形式存入数据库，这个过程即是[索引](http://baike.baidu.com/view/262241.htm)（index).
   2. 搜索[引擎](http://baike.baidu.com/view/53607.htm)的核心数据结构为[倒排文件](http://baike.baidu.com/view/228996.htm)（也称[倒排索引](http://baike.baidu.com/view/676861.htm)），
3. **搜索词处理**
   1. 用户在搜索[引擎](http://baike.baidu.com/view/53607.htm)界面输入关键词，单击“搜索”按钮后，搜索引擎程序即对[搜索词](http://baike.baidu.com/view/58654.htm)进行处理，
   2. 如中文特有的分词处理，去除[停止词](http://baike.baidu.com/view/2860648.htm)，判断是否需要启动整合搜索，
   3. 判断是否有拼写错误或错别字等情况。
4. 排序
   1. 对[搜索词](http://baike.baidu.com/view/58654.htm)处理后，搜索引擎程序便开始工作，从索引数据库中找出所有包含搜索词的[网页](http://baike.baidu.com/view/828.htm)，并且根据排名算法计算出哪些网页应该排在前面，然后按照一定格式返回到“搜索”页面。
   2. 再好的搜索引擎也无法与人相比，这就是为什么网站要进行[搜索引擎优化](http://baike.baidu.com/view/7147.htm)。没有[SEO](http://baike.baidu.com/view/1047.htm)的帮助，[搜索](http://baike.baidu.com/view/8638.htm)引擎常常并不能正确的返回最相关、最权威、最有用的[信息](http://baike.baidu.com/view/1527.htm)。
5. 简述信息检索过程中的主要技术有哪些，和搜索引擎比较有哪些差别？都在那些章里进行了讲述。

答:

**信息检索主要技术:**

* + - 1. 文本分析
    1. NLP

2. 建立索引

1. 查询
   * 1. 查询分析 NLP
     2. 相关度计算

和信息检索模型相关

1. 排序
   1. 实验室评价

**搜索引擎的主要技术:**

1. 爬虫 (第2讲 网络爬虫技术)
2. 文本分析
   1. NLP
3. 索引
4. 查询
   1. 查询分析 NLP
   2. 相关度计算 (第8讲 相似度计算)
5. 排序 (第9讲 检索排序)
6. 用户反馈
   1. refine query , relaxing query
7. 简述爬虫程序的工作过程

答:

1. 从一个或若干初始网页的[URL](http://baike.baidu.com/view/1496.htm)开始，
2. Fetch and parse them
   1. Extract URLs they point to
   2. Place the extracted URLs on a queue
3. Fetch each URL on the queue and repeat
   1. 直到满足系统的一定停止条件。
4. 简述网站防爬取策略有哪些？，对这些策略爬虫如何应对的

答:

网站防爬取措施：Robot 协议、IP 屏蔽、登录、JavaScript 渲染

* 1. Robot 协议：网站通过其告诉搜索引擎哪些页面可以抓取，哪些页面不能抓取。 Robot.txt 文件是一个文本文件。当一个搜索蜘蛛访问一个站点时，它会首先检查 该站点根目录下是否存在 robots.txt，如果存在，搜索机器人就会按照该文件中的 内容来确定访问的范围。
  2. IP 屏蔽：网站查看 Useragent，如果不是浏览器的，就封 IP；如果同一 IP 频繁 访问同一网站，同样封 IP。

应对措施：伪造 Useragent；连接代理服务器、多 IP 并行、增大爬取时间间隔。

* 1. 访问限制：交互登陆，提交用户名，口令。JavaScript 渲染,AJAX, Cookie，JSON。 动态网页，数据在后台数据库，通过 GET（POST)参数，后台 PHP 程序生成的网页。 应对措施：模拟浏览器工作(HTTP 分析工具可以分析 HTTP 传递的口令)
  2. 验证码

应对措施：脚本或人工对其图片进行爬虫遍历，然后将所有的图片保 存后与关键字进行对比并关联入库(分割验证码图像、丢进百度识图 API 函数、 返回百度识图结果)。

1. 简述正则表达式方法和DOM树分析方法的异同点

答:

正则表达式是对字符串操作的一种逻辑公式，就是用事先定义好的一些特定字 符、及这些特定字符的组合，组成一个“规则字符串”，这个“规则字符串”用来表达对字符串的一种过滤逻辑。

DOM 将 HTML 视为树状结构的元素，所有元素以及他们的文字和属性可通过 DOM 树来操作与访问。

**相同点:**

* 1. 都可用于网页内容的分析和信息的提取和替换

**不同点:**

* 1. 正则表达式匹配
     1. 正则表达式匹配速度快，但表达能力较弱，只具有正规文法的表示能力。
     2. 在对网页内容的信噪比要求不高的情况下可以使用基于正则表达式匹配的爬取程序
  2. HTML DOM树
     1. 提取HTML DOM树提取在解析HTML时速度较慢，但其表达能力相当于上下文无关文法。
     2. 在网页自动分类等需要进行网页去噪处理的情况时使用基于HTML DOM树的爬取程序。

1. 简述词干还原(Stemming)和词形归并(Lemmatization)的异同点

答:

词干还原通常指去除单词两端词缀的启发式过程。其能够提高召回率，但是会降低准确率。

词形归并利用词汇表和词分析来减少屈折变化的形式，将其转变为基本形式。其可以减少词项词典中的词项数量。

**异点：**

1)代表意义不同：Stemming 通常指很粗略的去除单词两端词缀的启发式过程。Lemmatization 通常指利用词汇表和词形分析来去除屈折词缀，从而返回词的原形或词典中的词的过程。

2)词干还原在一般情况下会将多个派生相关词合并在一起，而词形归并通常只将同一词元的不同屈折形式进行合并。

**同点：**

1)都体现了不同语言之间的差异性，包括：不同语言之间的差异，特殊专业语言与一般语言的差异。

2)词干还原或者词形归并往往通过在索引过程中增加插件程序的方式来实现

1. 建立词项词典的步骤？

答:

1. 文档解析
2. 词条化: 将给定的字符序列拆分成一系列子序列的过程，其中每一个子序列称之为一个“词条”Token。利用空格，标点符号进行分割.
3. 词项归一化: 将文档和查询中的词条“归一化”成一致的形式. 在IR系统的词项词典中，形成多个近似词项的一个等价类.
4. 词干还原: 去除单词两端词缀的启发式过程. 词干还原能够提高召回率，但是会降低准确率.
5. 词形归并: 利用词汇表和词形分析来减少屈折变化的形式，将其转变为基本形式。词形归并可以减少词项词典中的词项数量.
6. 停用词: 应用太广泛，对这样的词搜索引擎无法保证能够给出真正相关的搜索结果，难以帮助缩小搜索范围，同时还会降低搜索的效率
7. 什么是HMM？简述基于HMM的中文分词方法

答:

HMM，Hidden Markov Model，隐马尔可夫模型，是一个统计模型，用来描述一个含有隐含未知参数的马尔可夫过程。 可用于中文分词.

**HMM 是一个五元组：**

StatusSet(状态值集合)：输出的分词结果，状态值集合为(B, M, E, S)：{B:begin, M:middle, E:end, S:single}

ObservedSet(观察值集合)：输入的句子

TransProbMatrix(转移概率矩阵)：一个 4×4 的矩阵，表示不同状态之间转移的概率 EmitProbMatrix(发射概率矩阵)：表示在某一状态下对应到某字的概率

InitStatus(初始状态分布)：表示句子的第一个字属于{B, M, E, S}这四种状态的概率 Viterbi算法，一种动态规划算法，它用于寻找最有可能产生观测事件序列的维特比路径——隐含状态序列。对应于中文分词，它用来寻找最有可能产生某一句子的 BEMS 状态值序列。

1. 简述布尔模型模型及其特点

答:

布尔模型是一种简单的检索模型，建立在经典的集合论和布尔代数的基础上。遵循两条基本规则: 每个索引词在一篇文档中只有两种状态：出现或不出现， 对应权值为 0 或 1。

优点：1)查询简单，容易理解。2)通过使用复杂的布尔表达式，可方便地控制 查询结果 3)相当有效的实现方法 4)经过训练的用户可以容易地写出布尔查询式

缺点：1)是精确匹配，信息需求的能力表达不足，不能输出部分匹配的情况。 2)无权重设计，无法排序 3)用户必须会用布尔表达式提问，一般而言，检出的文 档或者太多或者太少。4)很难进行自动的相关反馈。

1. 简述向量空间模型及其特点

答:

向量空间模型：每篇文档表示成一个基于 tf-idf 权重的实值向量∈R|V|(V是词项集合，|V|表示词项个数)。文本内容的处理简化为向量空间中的向量，以空间上的相似度表达语义的相似度。

特点：1)维度非常高：特别是互联网搜索引擎，空间可能达到千万维或更高 2) 向量空间非常稀疏：对每个向量来说大部分都是 0

优点：1)帮助改善了检索结果 2)部分匹配的文档也可以被检索到 3)可以基于向 量 cosine 的值进行排序，提供给用户。

缺点：1)这种方法假设标记词是相互独立的，但实际可能不是这样，如同义词、近义词等往往被认为是不相关的词。

1. 简述非精确top K检索的策略

答:

找一个文档集合 A，K < |A| << N，利用 A 中的 top K 结果代替整个文档集的 top K 结果。即给定查询后，A 是整个文档集上近似剪枝得到的结果。

**策略一：索引去除(Index elimination)**，对于一个包含多个词项的查询来说， 很显然可以仅仅考虑那些至少包含一个查询词项的文档（进一步拓展思路：只考虑那些词项的 idf 值超过一定阈值的文档，只考虑包含多个查询词项）

**策略二：胜者表(Champion list)**，对于词典中的每个词项 t，预先计算出 r 个最高权重的文档。词项 t 所对应的 tf 值最高的 r 篇文档构成 t 的胜者表，也称为优胜表(fancy list)或高分文档(top doc)。其中 r 的值需要在索引建立时给定。

**策略三：静态得分**，希望排序靠前的文档既是相关的又是权威的。相关性通过余弦相似度得分来判断；权威性是与 query 无关的文档本身的属性决定的。

**策略四：影响度(Impact)排序**，多个 term 对应的文档次序不是统一的，即多种 顺序(文档内容相关的排序方式)。将词项 t 对应的所有文档 d 按照 tft,d值降序排列(不同的文档对不同的 t 具有不同的顺序)。

**策略五：簇剪枝方法——预处理**，随机选篇文档作为先导者。对于其它文档，计算和它最近的先导者。这些文档依附在一个先导者上面，称为追随者 (follower)。这样一个先导者平均大约有个追随者。给定查询 q，通过与先导 者计算余弦相似度，找出和它最近的一个先导者 L，候选集合 A 包括 L 及其追 随者，然后对 A 中的所有的文档计算余弦相似度。

1. 简述常用的网页排序算法

答:

**链接分析排序算法:**

**Pagerank算法:** 在随机游走过程中访问越频繁的网页越重要。基于“从许多优质的网页链接过来的网页，必定还是优质网页”的回归关系，来判定所有网页的重要性。



其中，PR(A)表示页面A的级别，页面Ti链向页面A，L(Ti) 是页面Ti 链出的链接数量, d取值在0到1之间，d也称为阻尼系数，由于用户不可能无限的单击下去，常常因劳累而随机跳入另一个页面. 1-d则是页面本身所具有的网页级别。

**HITS算法:** 超链导向的主题搜索(Hyperlink - Induced Topic Search)，对每个网页都要计算两个值: 权威值(authority)与中心值(hub)。

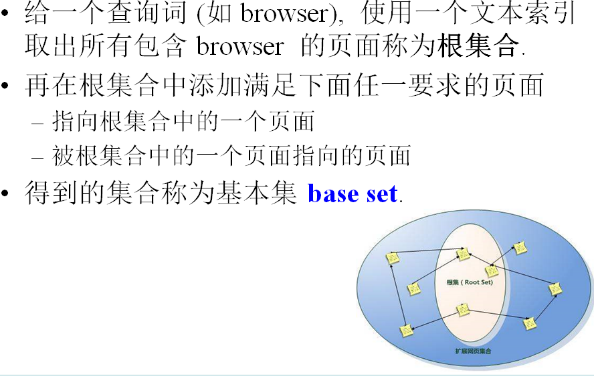
权威网页：一个网页被多次引用，则它可能是很重要的；一个网页虽然没有被多次引用，但是被重要的网页引用，则它也可能是很重要的；一个网页的重要性被平均的传递到它所引用的网页。

中心网页：提供指向权威网页的链接集合的 WEB 网页，它本身可能并不重要，或者说没有几个网页指向它，但是它提供了指向就某个主题而言最为重要的站点的链接集合.

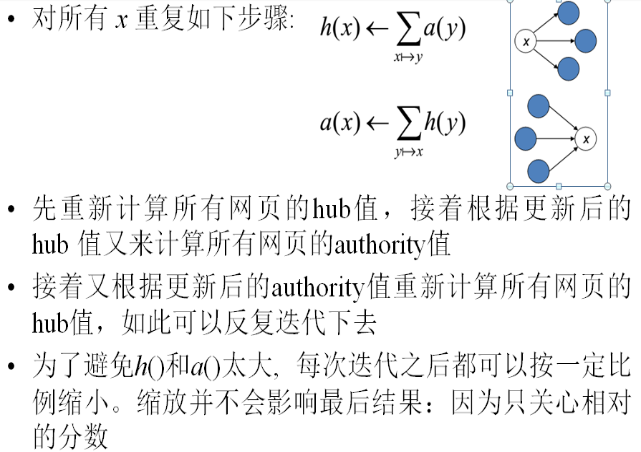
Hub页和Authority页之间是相互增强的关系.

算法步骤:

* + - 1. 确定基本集.



* + - 1. 对于基本集中的每一个页面x计算Hub分h(x)和Authority分a(x); 初始化: 所有的 x, h(x)=1; a(x)=1.
      2. 迭代计算



**Hilltop算法:** HillTop算法的指导思想和PageRank一致, 主题相关网页之间的链接对于权重计算的贡献比主题不相关的链接价值要更高。

Hilltop算法定义一个网站与其它网站的相关性，作为识别跨站点的链接交换干扰与识别相似链接的技术，以杜绝那些想通过任意链接来扰乱排名规则、那些想通过增加无效链接来提高网页PageRank值的做弊行为。

**Direct hit算法:** 注重信息的质量和用户反馈的排序方法。它的基本思想是:

1. 搜索引擎将查询的结果返回给用户，并跟踪用户在检索结果中的点击。
2. 如果返回结果中排名靠前的网页被用户点击后，浏览时间较短，用户又重新返回点击其它的检索结果，那么可以认为其相关度较差，系统将降低该网页的相关性。
3. 另一方面，如果网页被用户点击打开进行浏览，并且浏览的时间较长，那么该网页的受欢迎程度就高，相应地，系统将增加该网页的相关度。

在这种方法中，相关度在不停地变化，对于同一个词在不同的时间进行检索，得到结果集合的排序也有可能不同，它是一种动态排序。

**机器学习排序:** L2R**,** 有效利用日志数据与统计学习方法，使网页相关度与重要度计算的精度有了进一步的提升

1. 简述PageRank算法和HITS算法的异同点

答:

**同点：**都是基于链接分析的搜索引擎排序算法，并且在算法中两者都利用了特征向量作为理论基础和收敛性依据。

**异点：**HITS算法计算的authority值只是相对于某个检索主题的权重，因此HITS 算法也常被称为 Query-dependent 算法；而 PageRank 算法是独立于检索主题，因 此也常被称为 Query-independent 算法。

1. 简述信息检索的主要评价指标

答:

**查准率：**返回的结果中真正相关结果的比率

**查全率：**返回的相关结果数占实际相关结果总数的比率

**F 值：**查全率 R 和查准率 P 的加权调和平均值，F1 标准则综合了查准率和查全率，将两者赋予同样的重要性来考虑。

**R-查准率 (Precision@K (P@K), 这里的K就是R)：**计算序列中第 R 个位置文献的查准率, 即只考虑前R个文档时的查准率。R 是指与当前查询相关的文档总数。

(为了方便理解, 查看下面例子)

 R=10, R-查准率=4/10; R=3, R-查准率=2/3

**对多个查询进行查准率评估：**1)平均：宏平均(Macro Average): 对每个查询求出某个指标，然后对这些指标进行算术平均；微平均(Micro Average): 将所有查询视为一个查询，将各种情况的文档总数求和，然后进行指标的计算 2)查准率直方图：用于快速比较两个检索算法的性能。在多个查询下，分别计算多个检索算法在每一查询下的 R-查准率，计算其差值，并用直方图表示。

**查准率/查全率曲线：**在查全率和查准率间进行权衡。

a) 原始的曲线常常呈现锯齿状

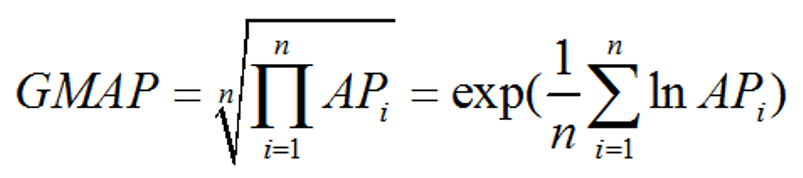
b) 需要去掉锯齿，进行平滑。采用插值查准率,记为PINTERP

c) 在查全率为R的位置的插值查准率，定义为查全率不小于R的位置上的查准率的最大值

**MAP (Mean Average Precision):** 平均查准率均值. 在每个相关文档位置上查准率的平均值被称为平均查准率 Average Precision (AP).

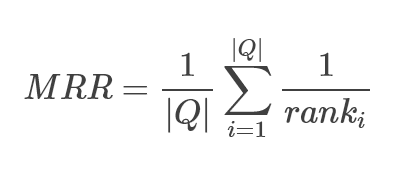
* 对于1次查询, 每个相关文档的rank如下
  + K1, K2, … KR
    - Compute Precision@K for each K1, K2, … KR
  + Average precision 1 = average of P@K
* 对于n次查询, 得到n个Average Precision, 则MAP = sum(Average Precision1到n)/n

**GMAP(Geometric MAP):** 几何平均查准率均值.



**MRR(Mean Reciprocal Rank):** reciprocal rank是指，第一个正确答案的排名的倒数。MRR是指多个查询语句的第一个正确答案排名倒数的均值.





**NDCG** **(Normalized discounted cumulative gain):** 一种总体观察检索排序效果的方法，过程如下:

对每个搜索结果根据相关度打分, 相关度大的文档比相关度小的文档最用户来说更重要.

* **Cumulative Gain (CG) 累积增益：**位于位置1 到p 的检索结果的相关度之和。



*reli* 表示第 i 个文档与查询语句的相关度。

特点：未考虑位置，即前p项中两文档交换不影响计算结果。

* **Discounted Cumulative Gain (DCG) :** 基本思想，若搜索算法把相关度高的文档排在后面，则应该给予惩罚。一般用log 函数表示这种惩罚。DCG 的计算如下：



* Ideal DCG (IDCG) : 将检索结果按照相关度从大到小排序, 然后在计算DCG, 公式和上面一样.
* Normalized Discounted Cumulative Gain (NDCG) : 公式表示如下:

**面向用户的测度方法：**

**覆盖率：**实际检出的相关文档中，用户已知的相关文档所占的比例。

**新颖性：**检出的相关文档中，用户未知的相关文档所占的比例。

**多样性：**检索结果的多样性，检出的相关文档中，不含语义上非常相似或近 似 copy 的文档

1. 什么是LSA，简述LSA的作用

答:

**LSA(Latent Semantic Analysis，隐语义分析)**：使用统计计算的方法对大量的文本集进行分析，从而提取出词与词之间潜在的语义结构，并用这种潜在的语义结构，来表示词和文本，达到消除词之间的相关性和简化文本向量实现降维的目的。

**基本观点是：**把高维的向量空间模型（VSM）表示中的文档映射到低维的潜在语义空间中。

**优势：**1)文章和单词都映射到同一个语义空间。2)语义空间的维度明显明显少于源单词-文章矩阵。

**应用：**1)在低维语义空间可对文档进行比较，进而可用于文档聚类和文档分类。 2)在翻译好的文档上进行训练，可以发现不同语言的相似文档，可用于跨语言检 索。3)发现词与词之间的关系，可用于同义词、歧义词检测。4)通过查询映射到 语义空间，可进行信息检索。5)从语义的角度发现词语的相关性，可用于“选择题回答模型”

1. 什么是PlSA，简述PlSA和LSA的异同点

答:

**PLSA(Probabilistic Latent Semantic Analysis，概率潜在语义分析)**，基于双模式和共现的数据分析方法延伸的经典的统计学方法，以统计学的角度来看待 LSA。 PLSA 中生成文档的整个过程便是选定文档生成主题，确定主题生成词。

**PLSA生成文档的过程**

* 1. pLSA中生成文档的整个过程便是选定文档生成主题，确定主题生成词。
  2. 自动地发现文档集中的主题（分布）
     1. 根据大量已知的文档-词项信息p(w|d) ，
     2. 训练出文档-主题p(z|d)和主题-词项p(w|z)

**异点：**LSA 隐含高斯分布假设，PLSA 隐含 Multi-nomial 分布假设；PLSA 的优化目标是是 KL-divergence 最小，LSA 依赖于最小均方误差(SVD是一种最小二乘法)等准则；PLSA 使用的 EM 算法需要反复迭代，这需要很大的计算量。

**同点：**LSA 和 PLSA 都可以用于文档聚类和文档分类，处理同义词；都是分析文档语料库，以便找到它的新的低维表示；都不能生成新文档的模型。(胡编乱造的)

1. 什么是LDA，简述LDA和PLSA的异同点

答:

LDA(Latent Dirichlet Allocation，隐含狄利克雷分布)，不再认为主题分布（各个主题在文档中出现的概率分布）和词分布（各个词语在某个主题下出现的概率 分布）是唯一确定的，而是有很多种可能，Dirichlet 先验为某篇文档随机抽取出 某个主题分布和词分布。

**同点：**LDA 和 PLSA 思想上一致：增加了 Dirichlet 先验，全贝叶斯化

**异点：**PLSA 认为文档 d 产生主题 z 的概率，主题 z 产生单词 w 的概率都是两个固定的值，LDA 不再认为主题分布和词分布是唯一确定的，而是有很多种可能，Dirichlet 先验为某篇文档随机抽取出某个主题分布和词分布。

1. 简述文本相似度量方法

答:

**1. 基于字符的方法：**

**最长公共子序列：**通过计算出两个字符串/序列之间的最长公共子序列， 并使用这个子序列的长度来反映两个字符串/序列之间的相似程度。 3.

**编辑距离：**指两个字串之间，由一个转成另一个所需的最少编辑操作次数

**扩展的编辑距离：**在思想上与编辑距离一样，只是除插入、删除和替换操作外，还支持相邻字符的交换 这样一个操作。

**Needleman-Wunsch Similarity：**对插入错误和删除错误赋予较高的惩罚分数

**Smith-Waterman Similarity：**是一个局部最优比对方法，它的目的是找出两个序列之间连续且相同的子序列。

**Jaro Similarity 和 Jaro-Winkler Similarity：**考虑两个字符串之间相同字符的 顺序位置和个数，适用于像人名这样的较短字符串之间的比较。

**Hamming Distance：**用于长度相同的序列之间的比较，思想非常简单，就 是逐位比较得到的不同次数。 基于项的方法： 余弦相似度：用向量空间中两个向量夹角的余弦值作为衡量两个个体间差

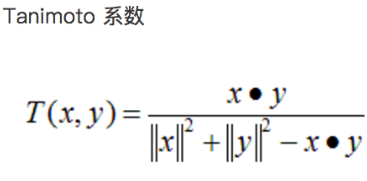
异的大小。

1. **基于term的方法**

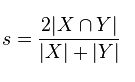
**Cosine Similarity:** 用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小。相比距离度量，余弦相似度更加注重两个向量在方向上的差异，而非距离或长度上。

**Jaccard Similarity：**两个集合 A 与 B 的交集的大小与 A 与 B 并集的大小的比值。

**Tanimoto系数:**  广义Jaccard相似度，元素的取值可以是实数。又叫作谷本系数. 如果我们的x,y都是二值向量，那么Tanimoto系数就等同Jaccard距离.



**Dice 系数：**Dice(s1,s2)=2×comm(s1,s2)/(len(s1)+len(s2))。即 2 倍的两个字符串相同字符的个数与两个字符串长度之和的比值。



2. 基于语料库的方法：通过对大量文档的统计分析得到语义上的相似

**语言模型：**假设具有相同(或相近)上下文的词，其语义是相近的。

**主题模型：**通过词与词的共现(Co-occurrence)来反映词与词之间的相似性

1. 简述常见的[LSH](https://baike.baidu.com/item/LSH)

答:

locality sensitive hash, LSH是指面对海量高维数据时，一般的算法无法快速降维查询相似度高的数据子集，利用特定的hash算法，将高维数据映射到低维空间，以较高概率快速寻找相似度高的数据子集。

**MinHash:** 可以用来快速估算两个集合的相似度。用于在搜索引擎中检测重复网页。集合A和B的相似度为集合A、B经过hash后最小哈希值相等的概率。

MinHash计算两个集合的相似度

* 第一种：使用多个hash函数
  + 可以选择一定数量的hash函数，比如K个。
  + 用这K个hash函数分别对集合A、B求哈希值，
    - 定义 Min(S)为集合S中具有最小哈希值的一个元素，
    - 对每个集合都得到K个最小值。比如Min(A)k={a1,a2,...,ak}，Min(B)k={b1,b2,...,bk}。
    - 集合A、B的相似度为Min(A)k和Min(B)k中相同元素个数与总的元素个数的比例。|Min(A)k ∩ Min(B)k| / |Min(A)k ∪ Min(B)k|
  + 计算复杂度高
* 第二种：使用单个hash函数
  + 定义hmink(S)为集合S中具有最小哈希值的K个元素。
  + 只需要对每个集合求一次哈希，然后取最小的K个元素。
  + 两个集合A、B的相似度，就是集合A中最小的K个元素与集合B中最小的K个元素的交集个数与并集个数的比例。

1. 什么是Simhash算法，简述其编码的实现步骤

答:

SimHash算法是LSH的一种, SIMHASH分为5步骤: 分词、HASH、加权、合并、降维.

* 1. 分词
  + 给定一段语句，进行分词，得到有效的特征向量，然后为每一个特征向量设置权重, 权重代表这个单词在整条语句中的重要程度，数字越大代表越重要。
    - 权重可以是这个词出现的次数。
* 2. hash
  + 选择simhash的位数
    - 综合考虑存储成本以及数据集的大小，比如说32位
  + 通过hash函数计算各个特征向量的hash值
  + hash值为二进制数01组成的n-bit签名。
    - 比如“CSDN”的hash值Hash(CSDN)为100101，“博客”的hash值Hash(博客)为“101011”。
* 加权
  + 在hash值的基础上，给所有特征向量进行加权，即W = Hash \* weight，遇到1则hash值和权值正相乘，遇到0则hash值和权值负相乘。
    - 例如给“CSDN”的hash值“100101”
    - 加权得到：W(CSDN) = 100101 4 = 4 -4 -4 4 -4 4，
    - 给“博客”的hash值“101011”加权得到：
    - W(博客)=101011 5 = 5 -5 5 -5 5 5
* 合并
  + 将上述各个特征向量的加权结果累加，变成只有一个序列串。
    - 例如“CSDN”的“4 -4 -4 4 -4 4”和“博客”的“5 -5 5 -5 5 5”进行累加，
    - 得到“4+5 -4+-5 -4+5 4+-5 -4+5 4+5”，
    - 得到“9 -9 1 -1 1”。
* 降维
  + 对于n-bit签名的累加结果，如果大于0则置1，否则置0，
  + 从而得到该语句的simhash值
    - 例如把上面计算出来的“9 -9 1 -1 1 9”降维
    - 得到的01串为：“1 0 1 0 1 1”，从而形成它们的simhash签名。

每篇文档得到SimHash签名值后，接着计算两个签名的海明距离即可。

1. //简述Web spam的常用技术 (不考)

答:

**针对基于相关性的排序策略的 spam 方法**：term spam。在预处理阶段可能形成的所谓“重要性”因素。顾名思义，既然是在预处理阶段形成的，就是和用户查询无关的。1)保证与查询词相关 2)提高与查询词相关度

**针对基于连接分析的排序策略的 spam 方法：**link spam。1)创造一个诱饵系统 (honey pot，中文是我乱翻的，建议考试时候使用英文)。2)参与链接交换 3)渗透 网络目录 4)在博客、未经审核的留言板、留言簿或维基上发布链接 5)购买过期域名

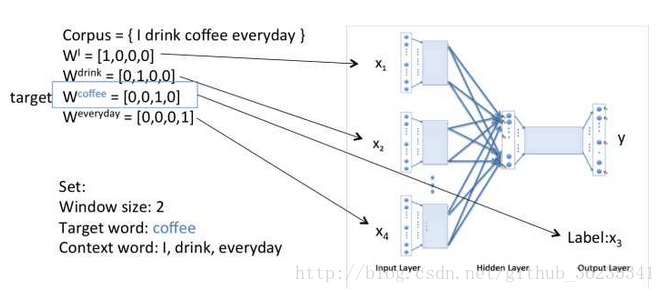
**Hiding techniques**：1)Terms，内容隐藏：与背景相同的字体，超小字体，单个 像素的图片做成的链接。2)Link，Cloaking：对机器人程序和普通用户返回不同 内容的页面。识别机器人程序：通过 IP 或者 user-agent；Redirection：载入网页 时自动转到另外一个 URL 地址：通过 refresh meta tag 或者 html 的脚本代码。

1. 简述CBOW（Continuous Bag of-Words）模型的训练过程

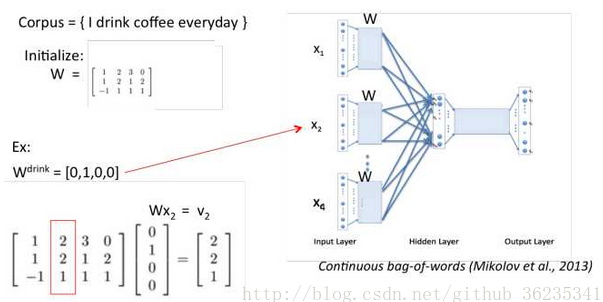
答:

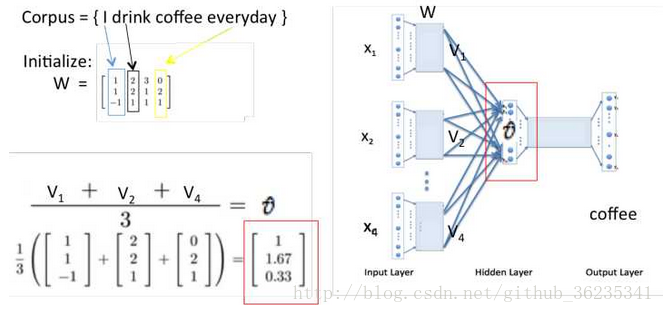
根据某个词前面的C个词或者前后C个连续的词，来计算某个词出现的概率。

* 假设 Corpus = { I drink coffee everyday } ，Target: “coffee”, 即根据”coffee”前C个和后C个单词来预测”coffee”出现的概率. C＝2时它的上下文分别“I”“drink”“everyday”. 模型输入：上下文的one hot表示
  + 1xV的向量, V词汇表大小

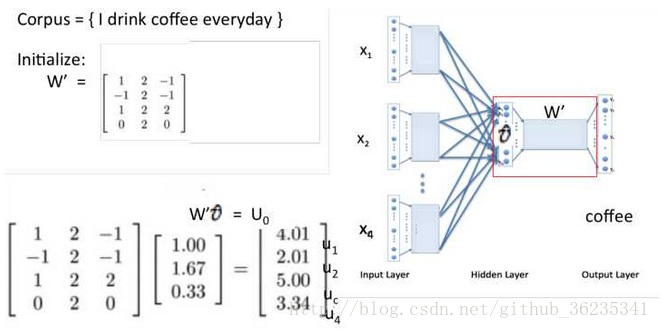


* 输入的每个One-hot分别跟同一个VxN的大小的系数矩阵W的相乘得到N个1xN的输出，然后N个取平均. (下图中的运算过程实际上是1xV乘VxN的转置, 不影响结果, 只是计算方式不同.)

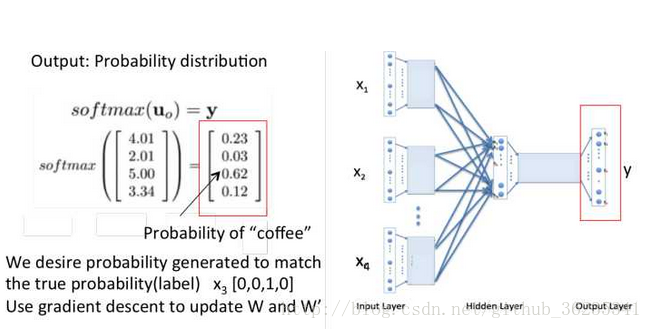




* 隐藏层跟另一个NxV大小的系数矩阵W’相乘得到1xV的输出层，这个输出层每个元素代表的就是词库里每个词的事后概率。



* 输出层需要跟ground truth也就是“coffee”的one hot形式做比较计算loss, 使用梯度下降更新W和W’，来最小化loss函数.



1. 简述Skip-gram模型的训练过程

答:

是根据某个词，然后分别计算它前后出现某几个词的各个概率。

1. 常见的局部形状的描述符有哪些？简述其思想

答:

**链码：**通过一个给定的方向上的单位尺寸的直线片段的序列来描述一条曲线或一个二维形状的边界。

**基于网格的方法：**将图像形状边界映射到一个标准的网格上，并将该形状边界调整到网格左上角，然后从左向右，从上到下扫描网格，若某个单元格被形状边界全部或者部分覆盖，则赋值 1，否则赋值 0，这样就得到了一个 0.1 组成的串，用来表征形状特征。

**距离直方图：**求得形状质心，在边界上均匀取特征点，计算特征点到质心距离， 建立起距离直方图。

**边界矩：**将边界点到质心的距离理解成一个分布，计算分布的矩。将低阶矩作 为特征。

**傅里叶描述子：**假定物体的形状是一条封闭的曲线，沿边界曲线上的一个动点 的坐标变化是一个以形状边界周长为周期的函数。这个周期函数可以展开成傅立 叶级数形式表示。傅立叶级数中的一系列系数 z(k)是直接与边界曲线的形状有关 的，称为傅立叶描述子。高频分量表示形状的细节，而低频分量则表示形状的总 体。当系数项取到足够阶次时，它可以将物体的形状信息完全提取并恢复出来。

**基于区域的形状描述符：**由两个阶段组成：由上而下的，通过对图像进行划分来发现同质的区域；由下而上地，对邻近同质的区域进行合并。

1. 什么是SIFT特征？简述SIFT特征的构建过程

答:

1. 什么是SIFT？
   1. 尺度不变特征转换，Scale-invariant feature transform
   2. 在空间尺度中寻找极值点，并提取出其位置、尺度、旋转不变量
2. SIFT算法大致分为四个步骤
   1. 步骤一：建立尺度空间
   2. 步骤二：在尺度空间中检测极值点，并进行精确定位和筛选
   3. 步骤三：特征点方向赋值，
   4. 步骤四：计算特征描述子
3. 简述将图像局部特征表示成全局特征的三种编码方法

答:

**BoW (Bag of visual word):** BoF (Bag of Features). 将图像视为一种文档对象, 图像中不同的局部区域或其特征可看做构成图像的词汇, 把每幅图像描述为一个局部区域/关键点(Patches/Key Points)特征的无序集合。使用某种聚类算法(如K-means)将局部特征进行聚类，每个聚类中心被看作是词典中的一个视觉词汇(Visual Word)，相当于文本检索中的词，

视觉词汇由聚类中心对应特征形成的码字(code word)来表示. 所有视觉词汇形成一个视觉词典, 图像中的每个特征都将被映射到视觉词典的某个词上，这种映射可以通过计算特征间的距离去实现, 然后统计每个视觉词的出现次数或频率.

**FV (Fisheries Vector):** 使用GMM混合高斯模型的聚类方法, 考虑了特征点到每个聚类中心的距离，也就是用所有聚类中心的线性组合去表示该特征点.

**VLAD (Aggregating local description):** 局部聚合向量. 可以认为VLAD是FV的简化版本, 只考虑离特征点最近的聚类中心

1. 推荐算法分类

答:

1. 根据推荐结果
   1. 大众化推荐: 搜索词联想推荐
   2. 个性化推荐: 需要了解用户喜好
2. 根据推荐方法
   1. 基于人口的统计学推荐: 根据系统用户的基本信息发现用户的相关程度，然后将相似用户喜爱的其他物品推荐给当前用户
   2. 基于内容的推荐: 将产品分解为一系列标签, 基于用户行为（浏览、购买、收藏）计算每个用户的产品兴趣标签, 针对所有新产品，分别计算每个用户的产品标签与每个新产品的相似度（基于cosine similarity）。
   3. 基于协同过滤的推荐 (Collaborative Filtering, CF): 是推荐系统中应用最早和最为成功的技术之一。收集用户的历史行为和偏好信息，利用群体智慧给出推荐

**item-based CF:**

通过用户对不同item的评分来评测item之间的相似性，基于item之间的相似性做出推荐；

预测目标用户对特定商品的喜好程度，系统从而根据这一喜好程度来对目标用户进行推荐。

**user-based CF:**

通过不同用户对item的评分来评测用户之间的相似性，基于用户之间的相似性做出推荐；

找到与此用户有相似兴趣的其他用户

* + 1. 用户相似度计算

然后将他们感兴趣的内容推荐给此用户。

* + 1. 邻居用户对商品评价的加权评价值
  1. 混合型推荐
  2. 基于社交网络关系的推荐

社交网络中的推荐问题可以分为两大类：

* 一类是利用用户的社交网络来给用户推荐朋友
  + 1. 它只需要社交网络用户的关系数据，即社交网络分析中的链接预测，也叫朋友推荐，
    2. 核心算法可参看社交网络分析中经典的链接预测算法，比如基于相似性的链路预测方法；
* 一类是利用社交信息来进行物品的推荐。
  + 1. 通过在传统的推荐算法的基础上融合社交边信息来提升用户的推荐性能，
    2. 它基于这样的假设：用户的偏好很容易受到它的社交朋友们的影响，即用户的偏好很大程度上与他的社交朋友类似。
  1. 基于模型的推荐: 依托于一些机器学习的模型，通过离线进行训练，在线进行推荐。与基于内存的推荐算法相对.
     1. 基于关联规则的推荐: 关联规则挖掘可以发现不同商品在销售过程中的相关性

1. 简述邻域协同过滤算法的思路和实现过程

答:

**思想:**

收集用户的历史行为和偏好信息，利用群体智慧给出推荐

**item-based CF:**

通过用户对不同item的评分来评测item之间的相似性，基于item之间的相似性做出推荐；

预测目标用户对特定商品的喜好程度，系统从而根据这一喜好程度来对目标用户进行推荐。

* 实现思路: 首先找到目标对象的最近邻居，由于当前用户对最近邻居的评分与对目标推荐对象的评分比较类似，所以可以根据当前用户对最近邻居的评分预测当前用户对目标推荐对象的评分，然后选择预测评分最高的若干个目标对象作为推荐结果呈现给当前用户

**user-based CF:**

通过不同用户对item的评分来评测用户之间的相似性，基于用户之间的相似性做出推荐；

找到与此用户有相似兴趣的其他用户

* + 1. 用户相似度计算

然后将他们感兴趣的内容推荐给此用户。

* + 1. 邻居用户对商品评价的加权评价值
* 实现思路: 首先找到和目标用户兴趣偏好相似的最近邻居，然后根据他的最近邻居对推荐对象的评分来预测目标用户对未评分的推荐对象的评分，选择预测评分最高的若干个推荐对象作为推荐结果反馈给用户
  + 使用皮尔森相关系数找到与当前用户最近邻的k个用户
  + 使用k个用户对当前用户尚未评分过的商品的评分预测当前用户对该商品的评分
  + 根据预测结果向当前用户推荐

1. 简述隐式因子模型

答:

隐式因子模型也就是矩阵分解模型, 包括SVD, FunkSVD, BiasSVD, SVD++等.

**SVD:** 奇异值分解. m个用户对n个商品的评分可以表示为m\*n矩阵, 矩阵中的值表示用户i对商品j的评分. 若用户还未对某个商品评分, 则其值为空. 由于可以通过SVD分解为, 如果我们要预测第i个用户对第j个物品的评分,则只需要计算即可.

**FunkSVD:** SVD将评分矩阵分解为三个矩阵的过程很耗时, 同时还面临稀疏问题, FunkSVD将评分矩阵分解为两个矩阵P和Q, 解决了上述问题. 训练时使用户的评分和用矩阵乘积得到的评分残差尽可能的小，也就是说，可以用均方差作为损失函数，来寻找最终的P 和Q 。

**BiasSVD:** FunkSVD的一种改进算法. BiasSVD假设评分系统包括三部分的偏置因素：一些和用户物品无关的评分因素，用户有一些和物品无关的评分因素，称为用户偏置项。而物品也有一些和用户无关的评分因素，称为物品偏置项。训练时在均方误差损失函数中考虑了用户偏置项和物品偏置项, 因此在某些场景中比FunkSVD效果好.

**SVD++:** SVD++在BiasSVD算法基础上进行了进一步加强, 增加了用户的隐式反馈.

* 用户对物品评分这种类型的数据为显式数据(explicit data)。
* 用户的行为数据，如浏览电影详情页，观看电影等，称之为隐式数据。
  + 隐式数据反应了用户的兴趣偏好，即隐式反馈。
  + 用户兴趣=显式兴趣+隐式反馈