



文本分类入门



主要内容

- 文本分类概述
- 文本文类的方法



何为文本分类？

- 文本/文档分类问题
 - 将一篇文档归入预先定义的几个类别中的一个或几个；
- 文本的自动分类：
 - 是使用计算机程序来实现这样的分类。
- 通俗点说，就好比 you 拿一篇文章，问计算机这篇文章要说的究竟是体育，经济还是教育，



两个前提

- 第一，用于分类所需要的类别体系是预先确定的。
 - 分类层次一旦确定，在相当长的时间内都是不可变的，或者即使要变更，也要付出相当大的代价（基本不亚于推倒并重建一个分类系统）。
- 第二，一篇文档并没有严格规定只能被分配给一个类别。
 - 这与分类这个问题的主观性有关，例如找10个人判断一篇文章所陈述的主题究竟属于金融，银行还是财政政策领域，10个人可能会给出不同的答案
 - 因此一篇文章很可能被分配到多个类别当中，只不过分给某些类别让人信服，而有些让人感觉模棱两可罢了（说的专业点，置信度不一样）。



广义的文本分类

- 文本分类可以判断一篇文章说的是什麼，可称之为“依据主题的分类”。
- 文本分类还可以：
 - 用于判断文章的写作风格；
 - 作者态度（积极？消极？）；
 - 甚至判断作者真伪（例如看看《红楼梦》最后二十回到底是不是曹雪芹写的）。
- 总而言之，凡是与文本有关，与分类有关，不管从什麼角度出发，依据的是何特征，都可以叫做文本分类。



文本分类 != 网页分类

- 网页所包含的信息远比含于其中的文字（文本）信息多得多；
- 对一个网页的分类，除了考虑文本内容的分类以外，链入链出的链接信息，页面文件本身的元数据，甚至是包含此网页的网站结构和主题，都能给分类提供莫大的帮助；
- 文本分类可看成是网页分类的一个子集。



文本分类的应用

- 搜索引擎;
- 数字图书馆;
- 档案管理;
-
- 凡跟海量文字信息打交道的系统，都用得上文本分类



Part 2. 文本分类的方法

文本分类问题与其它分类问题没有本质上的区别，其方法可以归结为：

- 根据待分类数据的某些特征来进行匹配，
- 当然完全的匹配是不太可能的，
- 因此必须（根据某种评价标准）选择最优的匹配结果，从而完成分类。



1. 核心问题-如何表示文本？

- 因此核心的问题便转化为用哪些特征表示一个文本才能保证有效和快速的分类。
- 对特征的不同选择主导着方法派别的不同。



2. 几种方法流派

- 基于词的匹配
- 基于知识规则
- 基于统计的方法（机器学习）
 - 人类的判断大多依据经验以及直觉，因此自然而然的会有人想到何让机器像人类一样自己来通过对大量同类文档的观察来自己总结经验，作为今后分类的依据。
 - 统计学习方法的基本思想（机器学习）



3. 简述基于机器学习的文本分类

■ 提供训练集：

- 统计学习方法需要一批由人工进行了准确分类的文档作为学习的材料（称为训练集，注意由人分类一批文档比从这些文档中总结出准确的规则成本要低得多）；

■ 挖掘规则，形成分类器：

- 计算机从这些文档重挖掘出一些能够有效分类的规则，这个过程被形象的称为训练，而总结出的规则集合常常被称为分类器。

■ 应用/测试

- 训练完成之后，需要对计算机从来没有见过的文档进行分类时，便使用这些分类器来进行。



Part 3 统计学习方法

统计学习方法进行文本分类就是让计算机自己来观察由人提供的训练文档集，自己总结出用于判别文档类别的规则和依据。理想的结果当然是让计算机在理解文章内容的基础上进行这样的分类，然而遗憾的是，我们所说的“理解”往往指的是文章的语义甚至是语用信息，这一类信息极其复杂，抽象，而且存在上下文相关性，对这类信息如何在计算机中表示都是尚未解决的问题（“知识表示”的问题）



1. 文本的表示

- **前提：**文档的内容与其中所包含的词有着必然的联系，同一类文档之间总存在多个共同的词，而不同类的文档所包含的词之间差异很大。
- 不光是包含哪些词很重要，**词频数**对分类也很重要，使用TF-IDF计算词频。

文本表示方法：

VSM的表示方法-词袋的表示方法

■ 向量模型（VSM，向量空间模型）是适合文本分类问题的文档表示模型。

- 在这种模型中，一篇文章被看作**单词特征项集合**来看，利用**加权特征项构成向量进行文本表示**；
- 利用**词频信息对文本特征进行加权**。它实现起来比较简单，并且分类准确度也高，能够满足一般应用的要求。

■ 缺点：

- 文本是一种信息载体，其所携带的信息由几部分组成：如组成元素本身的信息（词的信息）、组成元素之间顺序关系带来的信息以及上下文信息（更严格的说，还包括阅读者本身的背景和理解）
- VSM这种文档表示模型，基本上完全**忽略了除词的信息以外所有的部分**，这使得它能表达的信息量存在上限，也直接导致了基于这种模型构建的文本分类系统（虽然这是目前绝对主流的做法），几乎永远也不可能达到人类的分类能力。



文本表示方法： 潜在主题表示

- LSI
- PLSA
- LDA



2. 训练

计算机从给定的一堆文档中自己
学习分类的规则, 即得到分类器



VSM的表达方式举例

- $w1 = (\text{文本}, 5, \text{统计学习}, 4, \text{模型}, 0, \dots)$
 - 这个向量表示在 $w1$ 所代表的文本中，“文本”这个词出现了5次（这个信息就叫做词频），“统计学习”这个词出现了4次，而“模型”这个词出现了0次，依此类推，后面的词没有列出。
- 第2篇文章可以表示为
- $w2 = (\text{文本}, 9, \text{统计学习}, 4, \text{模型}, 10, \dots)$



数据词典的使用

- 只通过观察w2和w3我们就可以看出实际上有更方便的表示文本向量的方法，那就是把所有文档都要用到的词从向量中抽离出来，形成**共用的数据结构**（也可以仍是向量的形式），这个数据结构就叫做**词典**，或者**特征项集合**。
- 例如我们的问题就可以抽离出一个词典向量
 $D = (\text{文本}, \text{统计学习}, \text{模型}, \dots)$
- **所有的文档向量均可在参考这个词典向量的基础上，使用词频进行加权，文档则简化成诸如**
- $w1 = (5, 4, 0, \dots)$
- $w2 = (9, 4, 10, \dots)$



使用TF/IDF进行加权

- TF/IDF作为一个词对所属文档主题的贡献程度来说，是非常重要的度量标准，也是将文档转化为向量表示过程中的重要一环。
- 关于TF/IDF的详细解释，请看
http://googlechinablog.com/2006/06/blog-post_27.html



维数灾难问题

- 词典的维数巨大，引发维数灾难问题；
- 解决方法：
 - 方法1：先去停用词，再选取那些最具代表性的词汇（更严格的说法应该是，那些最具代表性的特征，为了便于理解，可以把特征暂时当成词汇来想象）。对这个问题的解决，有人叫它特征提取，也有人叫它降维。
 - 方法2：选用主题



朴素贝叶斯算法 (Naive Bayes)

- 贝叶斯算法关注的是文档属于某类别概率。
- 文档属于某个类别的概率等于文档中每个词属于该类别的概率的综合表达式。
- 而每个词属于该类别的概率又在一定程度上可以用这个词在该类别训练文档中出现的次数（词频信息）来粗略估计，因而使得整个计算过程成为可行的。
- 使用朴素贝叶斯算法时，在训练阶段的主要任务就是估计这些值。



使用朴素贝叶斯分类器

朴素贝叶斯分类器

朴素贝叶斯分类器是一种非常实用的方法。

1、数据特征

在朴素贝叶斯分类器中，每个实例 x 可由属性值的合取描述，而目标函数 $f(x)$ 从某有限集合 V 中取值。

2、贝叶斯方法

贝叶斯方法的新实例分类目标是在给定描述实例的属性值 $\langle a_1, a_2 \dots a_n \rangle$ 下，得到最可能的目标值 V_{MAP} 。

即：

$$v_{MAP} = \arg \max_{v_j} P(v_j \mid a_1, \dots, a_n)$$

$$v_{MAP} = \arg \max_{v_j \in V} \frac{P(a_1, a_2 \dots a_n \mid v_j) P(v_j)}{P(a_1, a_2 \dots a_n)} = \arg \max_{v_j \in V} P(a_1, a_2 \dots a_n \mid v_j) P(v_j)$$

上式中的有两个数据项需要估计：

(1) $P(v_j)$

常常是计算每个目标值 v_j 出现在训练数据中的频率。

(2) $P(a_1, \dots, a_n / v_j)$

除非有一个非常大的训练数据集，否则应用频率的方法无法获得可靠的估计。

3、朴素贝叶斯分类器 (NB)

朴素贝叶斯分类器的假定：在给定目标值下，属性值之间相互条件独立。换言之，给定实例的目标值情况下，观察到联合的 $a_1, a_2 \dots a_n$ 的概率正好是对每个单独属性的概率乘积：

$$P(a_1, \dots, a_n | v_j) = \prod_i P(a_i | v_j)$$

利用频率方法，从训练数据中估计不同 $P(a_i / v_j)$ 项的所需样本数比要估计 $P(a_1, \dots, a_n / v_j)$ 项所需的量小得多。

朴素贝叶斯分类器的定义：

$$v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

其中 v_{NB} 表示朴素贝叶斯分类器输出的目标值。

4、举例

提供了目标概念PlayTennis的14个训练样例，给新实例 $\langle \text{sunny}, \text{cool}, \text{high}, \text{strong} \rangle$ ，确定其分类。

根据表3-2， $V=\{\text{yes}, \text{no}\}$ ，可以计算出上式需要的概率值。

计算 $P(v_j)$

$$P(\text{yes})=9/14=0.64, \quad P(\text{no})=5/14=0.36$$

计算 $P(a_i/v_j)$

$$\text{例如: } P(\text{strong}/\text{yes})=3/9=0.33, \\ P(\text{strong}/\text{no})=3/5=0.60$$

计算 v_{NB}

$$v_{NB} = \arg \max_{v_j \in \{\text{yes}, \text{no}\}} P(v_j) \prod_i P(a_i | v_j) = \arg \max_{v_j \in \{\text{yes}, \text{no}\}} P(v_j) P(\text{sunny} | v_j) P(\text{cool} | v_j) P(\text{high} | v_j) P(\text{strong} | v_j)$$

$$P(\text{yes})P(\text{sunny}/\text{yes})P(\text{cool}/\text{yes})P(\text{high}/\text{yes})P(\text{strong}/\text{yes})=0.0053$$

$$P(\text{no})P(\text{sunny}/\text{no})P(\text{cool}/\text{no})P(\text{high}/\text{no})P(\text{strong}/\text{no})=0.0206$$

所以: $v_{NB}=\text{no}$



举例：学习分类文本

- 利用贝叶斯方法学习目标概念，然后用于文本自动过滤，比如
 - 我感兴趣的电子新闻稿
 - 讨论机器学习的万维网页
- 本节描述一个基于朴素贝叶斯分类器的文本分类的通用算法，它是目前所知的文本分类的最有效方法之一
- 问题框架：实例空间 X 包含了所有的文本文档，给定某未知目标函数 $f(x)$ 的一组训练样例， $f(x)$ 的值来自某有限集合 V （作为示例，此处令 $V=\{\text{like}, \text{dislike}\}$ ）



举例：学习分类文本（2）

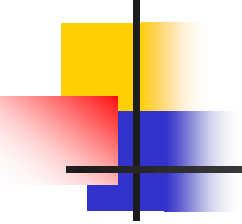
- 应用朴素贝叶斯分类器的两个主要设计问题：
 - 怎样将任意文档表示为属性值的形式
 - 如何估计朴素贝叶斯分类器所需的概率
- 表示文档的方法
 - 给定一个文本文档，对每个单词的位置定义一个属性，该属性的值为在此位置上找到的英文单词
- 假定我们共有1000个训练文档，其中700个分类为dislike，300个分类为like，现在要对下面的新文档进行分类：
 - This is an example document for the naive Bayes classifier. This document contains only one paragraph, or two sentences.

举例：学习分类文本（3）

■ 计算式

$$\begin{aligned}v_{NB} &= \arg \max_{v_j \in \{like, dislike\}} P(v_j) \prod_{i=1}^{19} P(a_i | v_j) \\&= \arg \max_{v_j \in \{like, dislike\}} P(v_j) P(a_1 = "this" | v_j) \dots P(a_{19} = "sentences" | v_j)\end{aligned}$$

- 注意此处贝叶斯分类器隐含的独立性假设并不成立。通常，某个位置上出现某个单词的概率与前后位置上出现的单词是相关的
- 虽然此处独立性假设不精确，但别无选择，否则要计算的概率项极为庞大。
- 另外实践中，朴素贝叶斯学习器在许多文本分类问题中性能非常好



$$\begin{aligned}
 v_{NB} &= \arg \max_{v_j \in \{like, dislike\}} P(v_j) \prod_{i=1}^{19} P(a_i | v_j) \\
 &= \arg \max_{v_j \in \{like, dislike\}} P(v_j) P(a_1 = "this" | v_j) \dots P(a_{19} = "sentences" | v_j)
 \end{aligned}$$

- 需要估计概率项 $P(v_i)$ 和 $P(a_i = w_k | v_i)$ 。前一项可基于每一类在训练数据中的比例很容易得到，后一项含三个参数，出现数据稀疏问题
- 再引入一个假定以减少需要估计的概率项的数量：假定单词 w_k 出现的概率独立于单词所在的位置，即

$$P(a_i = w_k | v_i) = P(w_k | v_j)$$

- 作此假定的一个主要优点在于：使可用于估计每个所需概率的样例数增加了，因此增加了估计的可靠程度
- 采纳 m -估计方法，即有统一的先验概率并且 m 等于词汇表的大小，因此

$$P(w_k | v_j) = \frac{n_k + 1}{n + |Vocabulary|}$$

用于学习和分类文本的朴素贝叶斯算法

■ Learn_Naive_Bayes_Text(Examples, V)

Examples为一组文本文档以及它们的目标值。V为所有可能目标值的集合。此函数作用是学习概率项 $P(w_k|v_j)$ 和 $P(v_j)$ 。

- 收集Examples中所有的单词、标点符号以及其他记号
 - Vocabulary ← 在Examples中任意文本文档中出现的所有单词及记号的集合
- 计算所需要的概率项 $P(v_j)$ 和 $P(w_k|v_j)$
 - 对V中每个目标值 v_j
 - $docs_j \leftarrow$ Examples中目标值为 v_j 的文档子集
 - $P(v_j) \leftarrow |docs_j| / |Examples|$
 - $Text_j \leftarrow$ 将 $docs_j$ 中所有成员连接起来建立的单个文档
 - $n \leftarrow$ 在 $Text_j$ 中不同单词位置的总数
 - 对Vocabulary中每个单词 w_k
 - $n_k \leftarrow$ 单词 w_k 出现在 $Text_j$ 中的次数
 - $P(w_k|v_j) \leftarrow (n_k+1) / (n+|Vocabulary|)$



用于学习和分类文本的朴素贝叶斯算法

- **Classify_Naive_Bayes_Text (Doc)**

对文档Doc返回其估计的目标值， a_i 代表在Doc中的第*i*个位置上出现的单词

- **positions** ← 在Doc中的所有单词位置，它包含能在Vocabulary中找到的记号

- 返回 v_{NB} ,
$$v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_{i \in positions} P(a_i | v_j)$$



实验结果

- Joachims将此算法用于新闻组文章的分类
 - 每一篇文章的分类是该文章所属的新闻组名称
 - 20个新闻组，每个新闻组有1000篇文章，共2万个文档
 - 2/3作为训练样例，1/3进行性能测量
 - 词汇表不包含最常用词（比如the、of）和罕见词（数据集中出现次数少于3）
- Lang用此算法学习目标概念“我感兴趣的新闻组文章”
 - NewsWeeder系统，让用户阅读新闻组文章并为其评分，然后使用这些评分的文章作为训练样例，来预测后续文章哪些是用户感兴趣的
 - 每天向用户展示前10%的自动评分文章，它建立的文章序列中包含的用户感兴趣的文章比通常高3~4倍

$$P(C_i | d) = \frac{P(d | C_i) P(C_i)}{P(d)}$$

$$P(d | C_i) = P(w_1 | C_i) P(w_2 | C_i) \dots P(w_i | C_i) \dots P(w_m | C_i)$$

$P(w_i | C_i)$ 就代表词汇 w_i 属于类别 C_i 的概率。

- 这其中就蕴含着朴素贝叶斯算法最大的两个缺陷。
 - 首先， $P(d | C_i)$ 之所以能展开成连乘积形式，就是假设一篇文章中的各个词之间是彼此独立的，其中一个词的出现丝毫不受另一个词的影响（回忆一下概率论中变量彼此独立的概念就可以知道），但这显然不对，即使不是语言学专家的我们也知道，词语之间有明显的所谓“共现”关系，在不同主题的文章中，可能共现的次数或频率有变化，但彼此间绝对谈不上独立。
 - 其二，使用某个词在某个类别训练文档中出现的次数来估计 $P(w_i | C_i)$ 时，只在训练样本数量非常多的情况下才比较准确，而需要大量样本的要求不仅给前期人工分类的工作带来更高要求（从而成本上升），在后期由计算机处理的时候也对存储和计算资源提出了更高的要求。



不同分类算法的效率

分类算法	召回率	准确率	F ₁ 测度
支持向量机(SVM)	80.2%	90.2%	84.9%
K 近邻算法(KNN)	82.3%	86.3%	84.3%
线性最小平方拟合算法 (LLSF)	84.2%	85.3%	84.8%
神经网络分类算法(NNet)	75.3%	79.8%	77.5%
朴素贝叶斯概率分类算法	73.4%	78.3%	75.8%



分类中的关键术语和概念解释

- 学习方法：使用样例（或称样本，训练集）来合成计算机程序的过程称为学习方法。
- 监督学习：学习过程中使用的样例是由输入/输出对给出时，称为监督学习。最典型的监督学习例子就是文本分类问题，训练集是一些已经明确分好了类别文档组成，文档就是输入，对应的类别就是输出。
- 非监督学习：学习过程中使用的样例不包含输入/输出对，学习的任务是理解数据产生的过程。典型的非监督学习例子是聚类，类别的数量，名称，事先全都没有确定，由计算机自己观察样例来总结得出。
- TSR (Term Space Reduction)：特征空间的压缩，即降维，也可以叫做特征提取。包括特征选择和特征抽取两大类方法。

- **分类状态得分** (CSV, Categorization Status Value): 用于描述将文档归于某个类别下有多大的可信度。
- **准确率 (Precision)**: 在所有被判断为正确的文档中, 有多大比例是确实正确的。
- **召回率 (Recall)**: 在所有确实正确的文档中, 有多大比例被我们判为正确。
- **假设**: 计算机对训练集背后的真实模型 (真实的分类规则) 的猜测称为假设。可以把真实的分类规则想像为一个目标函数, 我们的假设则是另一个函数, 假设函数在所有的训练数据上都得出与真实函数相同 (或足够接近) 的结果。
- **泛化性**: 一个假设能够正确分类训练集之外数据 (即新的, 未知的数据) 的能力称为该假设的泛化性。

- **一致假设：** 一个假设能够对所有训练数据正确分类，则称这个假设是一致的。
- **过拟合：** 为了得到一致假设而使假设变得过度复杂称为过拟合。想像某种学习算法产生了一个过拟合的分类器，这个分类器能够百分之百的正确分类样本数据（即再拿样本中的文档来给它，它绝对不会分错），但也就为了能够对样本完全正确的分类，使得它的构造如此精细复杂，规则如此严格，以至于任何与样本数据稍有不同文档它全都认为不属于这个类别！

- **超平面 (Hyper Plane)** : n 维空间中的线性函数唯一确定了一个超平面。一些较直观的例子, 在二维空间中, 一条直线就是一个超平面; 在三维空间中, 一个平面就是一个超平面。
- **线性可分和不可分**: 如果存在一个超平面能够正确分类训练数据, 并且这个程序保证收敛, 这种情况称为线形可分。如果这样的超平面不存在, 则称数据是线性不可分的。
- **正样本和负样本**: 对某个类别来说, 属于这个类别的样本文档称为正样本; 不属于这个类别的文档称为负样本。



文本分类之前的工作-预处理

从文本分类系统的处理流程来看，无论待分类的文本是中文还是英文，在训练阶段之前都要经过一个预处理的步骤，去除无用的信息，减少后续步骤的复杂度和计算负担。



中/英文文本分词

- 对中文文本来说，首先要经历一个分词的过程，就是把连续的文字流切分成一个一个单独的词汇（因为词汇将作为训练阶段“特征”的最基本单位），例如原文是“中华人民共和国今天成立了”的文本就要被切分成“中华/人民/共和国/今天/成立/了”这样的形式。
- 而对英文来说，没有这个步骤（更严格的说，并不是没有这个步骤，而是英文只需要通过空格和标点便很容易将一个个独立的词从原文中区分出来）。中文分词的效果对文本分类系统的表现影响很大，因为在后面的流程中，全都使用预处理之后的文本信息，不再参考原始文本，因此分词的效果不好，等同于引入了错误的训练数据。
- 分词本身也是一个值得大书特书的问题，目前比较常用的方法有词典法，隐马尔科夫模型和新兴的CRF方法。



能够下载中文分词包的网站

1) 斯坦福大学自然语言处理的网页，能够下载能够处理英文、中文等语义的分词工具。

<http://nlp.stanford.edu/software/tagger.shtml>

2) 中科院的能够处理分词的工具。

<http://ictclas.org/>

这个只能处理中文，有C++/java工具包可以下载。



举例

NO/x ./w 1/a 小/a 孔/n 用/p 一/m 包/q 绿豆/n 做/v 实验/vn , /w 其中/r 发芽/v 的/u 种子/n 有/v 100/m 粒/q , /w 没有

NO/x ./w 2/n 两/m 人/n 同/p 向/p 而/cc 行/v , /w 小红/nr 先/d 出发/v 。 /w 速度/n 是/v 12KM/x //w H/x , /w 20分钟/t

NO/x ./w 3/n 小/a 超/v 家/n 有/v 五/m 口/q 人/n , /w 爸爸/n 的/u 年龄/n 比/p 妈妈/n 大/a 2/n 岁/qt , /w 妈妈/n 的/

NO/x ./w 4/n 动物园/n 饲养员/n 每次/r 早餐/n 喂/v 北极熊/n 吃/v 3/n 只/q 企鹅/n , /w 午餐/n 喂/v 得/u 企鹅/n 数量/

NO/x ./w 5/f 一/m 辆/q 摩托车/n 上午/t 8时/t 从/p 甲地/n 出发/v , /w 以/p 每/r 小时/n 185千/m 米/q 的/u 速度/n 开/

NO/x 。 /w 6/g 一/m 对/p 热恋/v 的/u 情侣/n 落入/v 一个/mq 变态/n 杀人狂/n 手中/s , /w 面临/v 即将/d 双双/m 惨死/v

NO/x ./w 7/v 育/g 红/a 小学/n 5月/t 份/q 计划/n 用/p 电/n 480/m 度/qv , /w 实际/n 少/ad 用/v 60/m 度/qv , /w 请问/

NO/x ./w 8/a 一/m 件/q 衣服/n 打/v 八/m 折/q 出售/v 卖/v 100/m 元/q , /w 实际/ad 90/v 元/q 卖/v 出/v , /w 请问/v :

NO/x ./w 9/v 小/a 华/b 比/b 姐姐/n 小/a 12/m 岁/qt , /w 四年/m 后/f , /w 姐姐/n 的/u 年龄/n 刚好/d 是/v 小华/nr 的



去停止词

- 预处理中在分词之后的“去停止词”一步对两者来说是相同的，都是要把语言中一些表意能力很差的辅助性文字从原始文本中去除：
 - 对中文文本来说，类似“我们”，“在”，“了”，“的”这样的词汇都会被去除；
 - 英文中的“an”，“in”，“the”等也一样。这一步骤会参照一个被称为“停止词表”的数据（里面记录了应该被去除的词，有可能是以文件形式存储在硬盘上，也有可能是以数据结构形式放在内存中）来进行。



英文的词根还原

- 英文文本还有进一步简化和压缩的空间。我们都知道，英文中同一个词有所谓词形的变化（相对的，词义本身却并没有变），例如名词有单复数的变化，动词有时态的变化，形容词有比较级的变化等等，还包括这些变化形式的某种组合。而正因为词义本身没有变化，仅仅词形不同的词就不应该作为独立的词来存储和参与分类计算。去除这些词形不同，但词义相同的词，仅保留一个副本的步骤就称为“词根还原”，
- 例如在一篇英文文档中，经过词根还原后，“computer”，“compute”，“computing”，“computational”这些词全都被处理成“compute”（大小写转换也在这一步完成，当然，还要记下这些词的数目作为compute的词频信息）。



特征选取的客观标准

- 开方检验其实是数理统计中一种常用的检验两个变量独立性的方法.
- 开方检验最基本的思想就是通过观察实际值与理论值的偏差来确定理论的正确与否。

开方检验

- 开方检验最基本的思想就是通过观察实际值与理论值的偏差来确定理论的正确与否。
- 具体做的时候常常先假设两个变量确实是独立的（行话就叫做“原假设”），然后观察实际值（也可以叫做观察值）与理论值（这个理论值是指“如果两者确实独立”的情况下应该有的值）的偏差程度，如果偏差足够小，我们就认为误差是很自然的样本误差，是测量手段不够精确导致或者偶然发生的，两者确确实实是独立的，此时就接受原假设；如果偏差大到一定程度，使得这样的误差不太可能是偶然产生或者测量不精确所致，我们就认为两者实际上是相关的，即否定原假设，而接受悖论假设。

- 假设理论值为E（这也是数学期望的符号哦），实际值为x，如果仅仅使用所有样本的观察值与理论值的差值x-E之和

$$\sum_{i=1}^n (x_i - E) \qquad \sum_{i=1}^n (x_i - E)^2$$

$$\sum_{i=1}^n \frac{(x_i - E)^2}{E} \qquad (1)$$

当提供了数个样本的观察值 $x_1, x_2, \dots, x_i, \dots, x_n$ 之后，代入到式（1）中就可以求得开方值，用这个值与事先设定的阈值比较，如果大于阈值（即偏差很大），就认为原假设不成立，反之则认为原假设成立。

开方检验用于文本分类

- 在文本分类问题的特征选择阶段，我们主要关心一个词 t （一个随机变量）与一个类别 c （另一个随机变量）之间是否相互独立？
 - 如果独立，就可以说词 t 对类别 c 完全没有表征作用，即我们根本无法根据 t 出现与否来判断一篇文档是否属于 c 这个分类。
 - 但与最普通的开方检验不同，我们不需要设定阈值，因为很难说词 t 和类别 c 关联到什么程度才算是有表征作用，我们只想借用这个方法来选出一些最最相关的即可。
 - **原假设：词 t 与类别 C 不相关。**选择的过程也变成了为每个词计算它与类别 c 的开方值，从大到小排个序（此时开方值越大越相关），取前 k 个就可以。

■ 现在有 N 篇文档，其中有 M 篇是关于体育的，我们想考察一个词“篮球”与类别“体育”之间的相关性。我们有四个观察值可以使用：

1. 包含“篮球”且属于“体育”类别的文档数，命名为 A

2. 包含“篮球”但不属于“体育”类别的文档数，命名为 B

3. 不包含“篮球”但却属于“体育”类别的文档数，命名为 C

4. 既不包含“篮球”也不属于“体育”类别的文档数，命名为 D

特征选择	1. 属于 “体育”	2. 不属 于“体育”	总 计
1. 包含“篮球”	A	B	A+B
2. 不包含“篮球”	C	D	C+D
总 数	A+C	B+D	N

首先， **$A+B+C+D=N$** 。其次， **$A+C$** 的意思其实就是说“属于体育类的文章数量”，因此，它就等于 **M** ，同时， **$B+D$** 就等于 **$N-M$** 。

$$\frac{A+B}{N}$$

- 篮球出现的概率

$$E_{11} = (A+C) \frac{A+B}{N}$$

- 属于体育类别，且包含篮球的篇数

开方检验最基本的思想就是通过观察实际值与理论值的偏差来确定理论的正确与否。

$$\sum_{i=1}^n \frac{(x_i - E)^2}{E}$$

$$D_{11} = \frac{(A - E_{11})^2}{E_{11}}$$

$$\chi^2(\text{篮球}, \text{体育}) = D_{11} + D_{12} + D_{21} + D_{22}$$

把**D11**， **D12**， **D21**， **D22**的值分别代入并化简，可以得到

词**t**与类别**c**的开方值更一般的形式可以写成

$$\chi^2(t, c) = \frac{N(AD-BC)^2}{(A+C)(A+B)(B+D)(C+D)}$$

接下来我们就可以计算其他词如“排球”，“产品”，“银行”等等与体育类别的开方值，然后根据大小来排序，选择我们需要的最大的数个词汇作为特征项就可以了。



开发检验的后话

- 开方检验也并非就十全十美了。
- 想想A和B的值是怎么得出来的，它统计文档中是否出现词t，却不管t在该文档中出现了几次，这会使得他对低频词有所偏袒（因为它夸大了低频词的作用）。甚至会出现有些情况，一个词在一类文章的每篇文档中都只出现了一次，其开方值却大过了在该类文章99%的文档中出现了10次的词，其实后面的词才是更具代表性的，但只因为它出现的文档数比前面的词少了“1”，特征选择的时候就可能筛掉后面的词而保留了前者。这就是开方检验著名的“低频词缺陷”。因此开方检验也经常同其他因素如词频综合考虑来扬长避短。