文本分类入门

A Brief Introduction to Text Classification

王斌 收集 自

http://www.blogjava.net/zhenandaci/archive/2008/05/31/204646.html

作者：**Jasper** zhenandaci@msn.com

2009 年 12 月 12 日 于无锡

目 录

(一) 文本分类问题的定义

系列文章，从文本分类问题的定义开始，主要讲解文本分类系统的构成，主流的统计学习方法以及较为优秀的 SVM 算法及其改进。

一个文本(以下基本不区分“文本”和“文档”两个词的含义) 分类问题就是将一篇文档归入预先定义的几个类别中的一个或几个，而文本的自动分类则是使用计算机程序来实现这样的分类。通俗点说，就好比你拿一篇文章，问计算机这文章要说的究竟是体育，经济还是教育，计算机答不上就打它的屁屁(……) 。
注意这个定义当中着重强调的两个事实。
第一，用于分类所需要的类别体系是预先确定的。例如新浪新闻的分类体系，Yahoo!网页导航的分类层次。这种分类层次一旦确定，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 11 个不同的答案(聪明的读者，您应该能看出来并没有 11 个答案，这只是一种修辞方法，笑) ，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) 。

八股是一种写文章的格式，过去用于科举，现在用于科研，总之，和科学有点关系的文章就得八股，鉴于我正锻炼自己写论文的能力，所以按照标准的格式，陈述了文本分类问题的定义之后，我要说说它的应用范围。

现在一说到文本分类，大部分人想当然的将这个问题简化为判断一篇文章说的是什么，这只是文本分类的一小部分应用，我们可以称之为“依据主题的分类”。实际上，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，不管从什么角度出发，依据的是何特

征，都可以叫做文本分类。

当然，目前真正大量使用文本分类技术的，仍是依据文章主题的分类，而据此构建最多的系统，当属搜索引擎。内里的原因当然不言自明，我只是想给大家提个醒，文本分类还不完全等同于网页分类。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(比如新浪体育专栏里的网页毫无疑问都是关于体育的），因此说文本分类实际上是网页分类的一个子集也毫不为过。当然，纯粹的文本分类系统与网页分类也不是一点区别都没有。文本分类有个重要前提：即只能根据文章的文字内容进行分类，而不应借助诸如文件的编码格式，文章作者，发布日期等信息。而这些信息对网页来说常常是可用的，有时起到的作用还很巨大！因此纯粹的文本分类系统要想达到相当的分类效果，必须在本身的理论基础和技术含量上下功夫。

除了搜索引擎，诸如数字图书馆，档案管理等等要和海量文字信息打交道的系统，都用得上文本分类。另外，我的硕士论文也用得上(笑）。

下一章和大家侃侃与文本分类有关的具体方法概览，有事您说话。

（二）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。
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（注意这两方面的需求往往是互相矛盾的）。因此自有文本分类系统的那天起，就一直是对特征的不同选择主导着方法派别的不同。

最早的词匹配法仅仅根据文档中是否出现了与类名相同的词（顶多再加入同义词的处理）来判断文档是否属于某个类别。很显然，这种过于简单的方法无法带来良好的分类效果。

后来兴起过一段时间的知识工程的方法则借助于专业人员的帮助，为每个类别定义大量的推理规则，如果一篇文档能满足这些推理规则，则可以判定属于该类别。这里与特定规则的匹配程度成为了文本的特征。由于在系统中加入了人为判断的因素，准确度比词匹配法大为提高。但这种方法的缺点仍然明显，例如分类的质量严重依赖于这些规则的好坏，也就是依赖于制定规则的“人”的好坏；再比如制定规则的人都是专家级别，人力成本大幅上升常常令人难以承受；而知识工程最致命的弱点是完全不具备可推广性，一个针对金融领域构建的分类系统，如果要扩充到医疗或社会保险等相关领域，则除了完全推倒重来以外没有其他办法，常常造成巨大的知识和资金浪费。

后来人们意识到，究竟依据什么特征来判断文本应当隶属的类别这个问题，就连人类自己都不太回答得清楚，有太多所谓“只可意会，不能言传”的东西在里面。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。这便是统计学习方法的基本思想（也有人把这一大类方法称为机器学习，两种叫法只是涵盖范围大小有些区别，均无不妥）。
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），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。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。

现如今，统计学习方法已经成为了文本分类领域绝对的主流。主要的原因在于其中的很多技术拥有坚实的理论基础(相比之下，知识工程方法中专家的主观因素居多），存在明确的评价标准，以及实际表现良好。

下一章就深入统计学习方法，看看这种方法的前提，相关理论和具体实现。

(三) 统计学习方法

前文说到使用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(往大里说，这是一个“知识表示”的问题，完全可以另写一系列文章来说了），更不要说让计算机来理解。

利用计算机来解决问题的标准思路应该是：为这种问题寻找一种计算机可以理解的表示方法，或曰建立一个模型(一个文档表示模型）；然后基于这个模型，选择各方面满足要求的算法来解决。用谭浩强的话说，程序，就是数据+算法。(啥？你不知道谭浩强是谁？上过学么？学过 C 么？这捣什么乱？）

既然文本的语义和语用信息很难转换成计算机能够理解的表示形式，接下来顺理成章的，人们开始用文章中所包含的较低级别的词汇信息来表示文档，一试之下，效果居然还不错。

统计学习方法进行文本分类(以下就简称为“统计学习方法”，虽然这个方法也可以应用到除文本分类以外的多个领域）的一个重要前提由此产生，那就是认为：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[1]。

进一步的，不光是包含哪些词很重要，这些词出现的次数对分类也很重要。

这一前提使得向量模型(俗称的 VSM，向量空间模型）成了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。[5]

而实际上，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）[12]。

而 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[12]，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。后面我们也会谈到，相比于所谓的分类算法，对特征的选择，也就是使用哪些特征来代表一篇文档，往往更能影响分类的效果。

对于扩充文档表示模型所包含的信息量，人们也做过有益的尝试，例如被称为LSI(Latent Semantic Index 潜在语义索引）的方法，就被实验证明保留了一定的语义信息(之所以说被实验证明了，是因为人们还无法在形式上严格地证明它确实保留了语义信息，而且这种语义信息并非以人可以理解的方式被保留下来），此为后话。

前文说到(就不能不用这种老旧的说法？换换新的，比如Previously on "Prison Break"，噢，不对，是Previously on Text Categorizaiton……）统计学习方法其实就是一个两阶段的解决方案，(1）训练阶段，由计算机来总结分类的规则；(2）分类阶段，给计算机一些它从来没见过的文档，让它分类(分不对就打屁屁）。

下一章就专门说说训练阶段的二三事。

(四) 训练Part 1

训练，顾名思义，就是 training(汗，这解释），简单的说就是让计算机从给定的一堆文档中自己学习分类的规则(如果学不对的话，还要，打屁屁？）。

开始训练之前，再多说几句关于 VSM 这种文档表示模型的话。
举个例子，假设说把我正在写的“文本分类入门”系列文章的第二篇抽出来当作一个需要分类的文本，则可以用如下的向量来表示这个文本，以便于计算机理解和处理。

w2=(文本，5，统计学习，4，模型，0，……)

这个向量表示在 w2 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。
而系列的第三篇文章可以表示为

w3=(文本，9，统计学习，4，模型，10，……)

其含义同上。如果还有更多的文档需要表示，我们都可以使用这种方式。
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=(文本，统计学习，模型，……)

所有的文档向量均可在参考这个词典向量的基础上简化成诸如

w2=(5，4，0，……)

w3=(9，4，10，……)

的形式，其含义没有改变。

5，4，10 这些数字分别叫做各个词在某个文档中的权重，实际上单单使用词频作为权重并不多见，也不十分有用，更常见的做法是使用地球人都知道的 TF/IDF 值作为权重。(关于 TF/IDF 的详细解释，Google 的吴军研究员写了非常通俗易懂的文章，发布于 Google 黑板报，链接地址是 http://googlechinablog.com/2006/06/blog-post_27.html，有兴趣不妨一读) 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。

在这个转化过程中隐含了一个很严重的问题。注意看看词典向量 D，你觉得它会有多大？或者说，你觉得它会包含多少个词？

假设我们的系统仅仅处理汉语文本，如果不做任何处理，这个词典向量会包含汉语中所有的词汇，我手头有一本商务印书馆出版的《现代汉语词典》第 5 版(2005 年 5 月出版)，其中收录了 65，000 个词，D 大致也应该有这么大，也就是说，D 是一个 65，000 维的向量，而所有的文本向量 w2,w3,wn 也全都是 65，000 维的！(这是文本分类这一问题本身的一个特性，称为“高维性”) 想一想，大部分文章仅仅千余字，包含的词至多几百，为了表示这样一个文本，却要使用 65，000 维的向量，这是对存储资源和计算能力多大的浪费呀！(这又是文本分类问题的另一个特性，称为“向量稀疏性”，后面会专门有一章讨论这些特性，并指出解决的方法，至少是努力的方向)

中国是一个人口众多而资源稀少的国家，我们不提倡一味发展粗放型的经济，我们所需要的可持续发展是指资源消耗少，生产效率高，环境污染少……跑题了……

这么多的词汇当中，诸如“体育”，“经济”，“金融”，“处理器”等等，都是极其能够代表文章主题的，但另外很多词，像“我们”，“在”，“事情”，“里面”等等，在任何主题的文章中都很常见，根本无法指望通过这些词来对文本类别的归属作个判断。这一事实首先引发了对文本进行被称为“去停止词”的预处理步骤(对英文来说还有词根还原，但这些与训练阶段无关，不赘述，会在

以后讲述中英文文本分类方法区别的章节中讨论），与此同时，我们也从词典向量 D 中把这些词去掉。

但经过停止词处理后剩下的词汇仍然太多，使用了太多的特征来表示文本，就是常说的特征集过大，不仅耗费计算资源，也因为会引起“过拟合问题”而影响分类效果[22]。

这个问题是训练阶段要解决的第一个问题，即如何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。特征提取实际上有两大类方法。一类称为特征选择(Term Selection)，指的是从原有的特征(那许多有用无用混在一起的词汇）中提取出少量的，具有代表性的特征，但特征的类型没有变化(原来是一堆词，特征提取后仍是一堆词，数量大大减少了而已）。另一类称为特征抽取(Term Extraction）的方法则有所不同，它从原有的特征中重构出新的特征(原来是一堆词，重构后变成了别的，例如 LSI 将其转为矩阵，文档生成模型将其转化为某个概率分布的一些参数），新的特征具有更强的代表性，并耗费更少的计算资源。(特征提取的各种算法会有专门章节讨论)

训练阶段，计算机根据训练集中的文档，使用特征提取找出最具代表性的词典向量(仍然是不太严格的说法），然后参照这个词典向量把这些训练集文档转化为向量表示，之后的所有运算便都使用这些向量进行，不再理会原始的文本形式的文档了(换言之，失宠了，后后）。

下一章继续训练，咱们之间还没完。(怎么听着像要找人寻仇似的)

(五) 训练Part 2

将样本数据成功转化为向量表示之后，计算机才算开始真正意义上的“学习”过程。

再重复一次，所谓样本，也叫训练数据，是由人工进行分类处理过的文档集合，计算机认为这些数据的分类是绝对正确的，可以信赖的(但某些方法也有针对训练数据可能有错误而应对的措施）。接下来的一步便是由计算机来观察这些训练数据的特点，来猜测一个可能的分类规则(这个分类规则也可以叫做分类器，在机器学习的理论著作中也叫做一个“假设”，因为毕竟是对真实分类规则的一个猜测），一旦这个分类满足一些条件，我们就认为这个分类规则大致正确并且足够好了，便成为训练阶段的最终产品——分类器！再遇到新的，计算机没有见过的文档时，便使用这个分类器来判断新文档的类别。

举一个现实中的例子，人们评价一辆车是否是“好车”的时候，可以看作一个分类问题。我们也可以把一辆车的所有特征提取出来转化为向量形式。在这个问题中词典向量可以为：

D=(价格，最高时速，外观得分，性价比，稀有程度)

则一辆保时捷的向量表示就可以写成

vp=(200 万，320，9.5，3，9)

而一辆丰田花冠则可以写成

vt=(15 万，220，6.0，8，3)

找不同的人来评价哪辆车算好车，很可能会得出不同的结论。务实的人认为性价比才是评判的指标，他会认为丰田花冠是好车而保时捷不是；喜欢奢华的有钱人可能以稀有程度来评判，得出相反的结论；喜欢综合考量的人很可能把各项指标都加权考虑之后才下结论。

可见，对同一个分类问题，用同样的表示形式(同样的文档模型），但因为关注数据不同方面的特性而可能得到不同的结论。这种对文档数据不同方面侧重的不同导致了原理和实现方式都不尽相同的多种方法，每种方法也都对文本分类这个问题本身作了一些有利于自身的假设和简化，这些假设又接下来影响着依据这些方法而得到的分类器最终的表现，可谓环环相连，丝丝入扣，冥冥之中自有天意

呀(这都什么词儿……)。

比较常见，家喻户晓，常年被评为国家免检产品(？！)的分类算法有一大堆，什么决策树，Rocchio，朴素贝叶斯，神经网络，支持向量机，线性最小平方拟合，kNN，遗传算法，最大熵，Generalized Instance Set 等等等等(这张单子还可以继续列下去)。在这里只挑几个最具代表性的算法侃一侃。

Rocchio 算法

Rocchio 算法应该算是人们思考文本分类问题时最先能想到，也最符合直觉的解决方法。基本的思路是把一个类别里的样本文档各项取个平均值(例如把所有“体育”类文档中词汇“篮球”出现的次数取个平均值，再把“裁判”取个平均值，依次做下去)，可以得到一个新的向量，形象的称之为“质心”，质心就成了这个类别最具代表性的向量表示。再有新文档需要判断的时候，比较新文档和质心有多么相像(八股点说，判断他们之间的距离)就可以确定新文档属不属于这个类。稍微改进一点的 Rocchio 算法不尽考虑属于这个类别的文档(称为正样本)，也考虑不属于这个类别的文档数据(称为负样本)，计算出来的质心尽量靠近正样本同时尽量远离负样本。Rocchio 算法做了两个很致命的假设，使得它的性能出奇的差。一是它认为一个类别的文档仅仅聚集在一个质心的周围，实际情况往往不是如此(这样的数据称为线性不可分的)；二是它假设训练数据是绝对正确的，因为它没有任何定量衡量样本是否含有噪声的机制，因而也就对错误数据毫无抵抗力。

不过 Rocchio 产生的分类器很直观，很容易被人类理解，算法也简单，还是有一定的利用价值的(做汉奸状)，常常被用来做科研中比较不同算法优劣的基线系统(Base Line)。

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。

朴素贝叶斯算法的公式只有一个

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

其中 P(d| Ci) =P(w1|Ci) P(w2|Ci) …P(wi|Ci) P(w1|Ci) …P(wm|Ci) (式 1)

P(wi|Ci) 就代表词汇 wi 属于类别 Ci 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| Ci) 之所以能展开成(式 1) 的连乘积形式，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(wi|Ci) 时，只在训练样本数量非常多的情况下才比较准确(考虑扔硬币的问题，得通过大量观察才能基本得出正反面出现的概率都是二分之一的结论，观察次数太少时很可能得到错误的答案) ，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。

kNN 算法则又有所不同，在 kNN 算法看来，训练样本就代表了类别的准确信息(因此此算法产生的分类器也叫做“基于实例”的分类器) ，而不管样本是使用什么特征表示的。其基本思想是在给定新文档后，计算新文档特征向量和训练文档集中各个文档的向量的相似度，得到 K 篇与该新文档距离最近最相似的文档，根据这 K 篇文档所属的类别判定新文档所属的类别(注意这也意味着 kNN 算法根本没有真正意义上的“训练”阶段) 。这种判断方法很好的克服了 Rocchio 算法中无法处理线性不可分问题的缺陷，也很适用于分类标准随时会产生变化的需求(只要删除旧训练文档，添加新训练文档，就改变了分类的准则) 。

kNN 唯一的也可以说最致命的缺点就是判断一篇新文档的类别时，需要把它与现存的所有训练文档全都比较一遍，这个计算代价并不是每个系统都能够承受的

（比如我将要构建的一个文本分类系统，上万个类，每个类即便只有 20 个训练样本，为了判断一个新文档的类别，也要做 20 万次的向量比较！）。一些基于 kNN 的改良方法比如 Generalized Instance Set 就在试图解决这个问题。

下一节继续讲和训练阶段有关的话题，包括概述已知性能最好的 SVM 算法。明儿见！（北京人儿，呵呵）

(六) 训练Part 3

SVM 算法

支持向量机(Support Vector Machine) 是Cortes 和Vapnik 于1995 年首先提出的，它在解决小样本、非线性及高维模式识别中表现出许多特有的优势，并能够推广应用到函数拟合等其他机器学习问题中[10]。

支持向量机方法是建立在统计学习理论的VC 维理论和结构风险最小原理基础上的，根据有限的样本信息在模型的复杂性(即对特定训练样本的学习精度，Accuracy) 和学习能力(即无错误地识别任意样本的能力) 之间寻求最佳折衷，以期获得最好的推广能力[14](或称泛化能力) 。

SVM 方法有很坚实的理论基础，SVM 训练的本质是解决一个二次规划问题(Quadruple Programming，指目标函数为二次函数，约束条件为线性约束的最优化问题) ，得到的是全局最优解，这使它有着其他统计学习技术难以比拟的优越性。SVM 分类器的文本分类效果很好，是最好的分类器之一。同时使用核函数将原始的样本空间向高维空间进行变换，能够解决原始样本线性不可分的问题。其缺点是核函数的选择缺乏指导，难以针对具体问题选择最佳的核函数；另外 SVM 训练速度极大地受到训练集规模的影响，计算开销比较大，针对 SVM 的训练速度问题，研究者提出了很多改进方法，包括Chunking 方法、Osuna 算法、SMO 算法和交互SVM 等等[14]。

SVM 分类器的优点在于通用性较好，且分类精度高、分类速度快、分类速度与训练样本个数无关，在查准和查全率方面都优于kNN 及朴素贝叶斯方法[8]。

与其它算法相比，SVM 算法的理论基础较为复杂，但应用前景很广，我打算专门写一个系列的文章，详细的讨论SVM 算法，stay tuned!

介绍过了几个很具代表性的算法之后，不妨用国内外的几组实验数据来比较一下他们的优劣。

在中文语料上的试验，文献[6]使用了复旦大学自然语言处理实验室提供的基准语料对当前的基于词向量空间文本模型的几种分类算法进行了测试，这一基准语料分为20 个类别，共有9804 篇训练文档，以及9833 篇测试文档。在经过统一的分词处理、噪声词消除等预处理之后，各个分类方法的性能指标如下。

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%

其中 F1 测度是一种综合了查准率与召回率的指标，只有当两个值均比较大的时候，对应的 F1 测度才比较大，因此是比单一的查准或召回率更加具有代表性的指标。

由比较结果不难看出，SVM 和 kNN 明显优于朴素贝叶斯方法(但他们也都优于 Rocchio 方法，这种方法已经很少再参加评测了）。

在英文语料上，路透社的 Reuters-21578 “ModApt´e”是比较常用的测试集，在这个测试集上的测试由很多人做过，Sebastiani 在文献[23]中做了总结，相关算法的结果摘录如下：

分类算法	在 Reuters-21578 “ModApt´e”上的 F1 测度
Rocchio	0.776
朴素贝叶斯	0.795
kNN	0.823
SVM	0.864

仅以 F1 测度来看，kNN 是相当接近 SVM 算法的，但 F1 只反映了分类效果(即分类分得准不准），而没有考虑性能(即分类分得快不快）。综合而论，SVM 是效果和性能均不错的算法。

前面也提到过，训练阶段的最终产物就是分类器，分类阶段仅仅是使用这些分类器对新来的文档分类而已，没有过多可说的东西。

下一章节是对到目前为止出现过的概念的列表及简单的解释，也会引入一些后面会用到的概念。再之后会谈及分类问题本身的分类(绕口），中英文分类问题的相似与不同之处以及几种特征提取算法的概述和比较，路漫漫……

(七) 相关概念总结

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[22]。

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[22]。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 [22]。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[22]。

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[22]。

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[22]。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的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[22]。

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。

规划：对于目标函数，等式或不等式约束都是线性函数的问题称为线性规划问题。对于目标函数是二次的，而约束都是线性函数的最优化问题称为二次规划问题[22]。

对偶问题：

给定一个带约束的优化问题

目标函数：$\min\ f(x)$

约束条件：$C(x) \geq 0$

可以通过拉格朗日乘子构造拉格朗日函数

$$L(x,\lambda) = f(x) - \lambda^T C(x)$$

令 $g(\lambda) = f(x) - \lambda^T C(x)$

则原问题可以转化为

目标函数：$\max g(\lambda)$

约束条件：$\lambda \geq 0$

这个新的优化问题就称为原问题的对偶问题(两个问题在取得最优解时达到的条件相同）。

(八) 中英文文本分类的异同

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。

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一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。

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。

对中文文本来说，到此就已初审合格，可以参加训练了(笑)。而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）。

经过预处理步骤之后，原始文档转换成了非常节省资源，也便于计算的形式，后面的训练阶段大同小异(仅仅抽取出的特征不同而已，毕竟，一个是中文词汇的集合，一个是英文词汇的集合嘛）。

下一章节侃侃分类问题本身的分类。

(九) 文本分类问题的分类

开始之前首先说说分类体系。回忆一下，分类体系是指事先确定的类别的层次结构以及文档与这些类别间的关系。

其中包含着两方面的内容：

一，类别之间的关系。一般来说类别之间的关系都是可以表示成树形结构，这意味着一个类有多个子类，而一个子类唯一的属于一个父类。这种类别体系很常用，却并不代表它在现实世界中也是符合常识的，举个例子，“临床心理学”这个类别应该即属于“临床医学”的范畴，同时也属于“心理学”，但在分类系统中却不便于使用这样的结构。想象一下，这相当于类别的层次结构是一个有环图，无论遍历还是今后类别的合并，比较，都会带来无数的麻烦。

二，文档与类别间的关系。一般来说，在分类系统中，我们倾向于让一篇文档唯一的属于一个类别(更严格的说，是在同一层次中仅属于一个类别，因为属于一个类别的时候，显然也属于这个类别的父类别），这使得我们只适用一个标签就可以标记这个文档的类别，而一旦允许文档属于多个类别，标签的数目便成为大小不定的变量，难于设计成高效的数据结构。这种“属于多个”类的想法更糟的地方在于文档类别表示的语义方面，试想，如果姚明给灾区捐款的新闻即属于灾区新闻，也属于体育新闻的话(这在现实中倒确实是合情合理的），当用户使用这个系统来查找文档，指定的条件是要所有“属于灾区新闻但不属于体育新闻的新闻”(有点拗口，不过正好练嘴皮子啦，笑）的时候，这篇姚明的报道是否应该包含在查询结果中呢？这是一个矛盾的问题。

文本分类问题牵涉到如此多的主题，本身又含有如此多的属性，因此可以从多个角度对文本分类问题本身进行一下分类。

分类系统使用何种分类算法是分类系统的核心属性。如果一个分类算法在一次分类判断时，仅仅输出一个真假值用来表示待分类的文档是否属于当前类别的话，这样的系统就可以叫做基于二元分类器的分类系统。有些分类算法天然就是独立二元的，例如支持向量机，它只能回答这个文档是或不是这个类别的。这种分类算法也常常被称为“硬分类”的算法(Hard Categorization) 。而有的算法在一

次判断后就可以输出文档属于多个类别的得分(假设说，得分越大，则说明越有可能属于这个类别），这类算法称为“排序分类”的算法(Ranking Categorization），也叫做 m 元分类算法。kNN 就是典型的 m 元分类算法(因为 kNN 会找出与待分类文档最相近的训练样本，并记录下这些样本所属的分类）。

(十) 特征选择算法之开方检验

前文提到过，除了分类算法以外，为分类文本作处理的特征提取算法也对最终效果有巨大影响，而特征提取算法又分为特征选择和特征抽取两大类，其中特征选择算法有互信息，文档频率，信息增益，开方检验等等十数种，这次先介绍特征选择算法中效果比较好的开方检验方法。

大家应该还记得，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。(什么？你是文史类专业的学生，没有学过数理统计？那你做什么文本分类？在这捣什么乱？)

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，而接受备择假设。

那么用什么来衡量偏差程度呢？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

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

来衡量，单个的观察值还好说，当有多个观察值 x_1，x_2，x_3 的时候，很可能 x_1-E，x_2-E，x_3-E 的值有正有负，因而互相抵消，使得最终的结果看上好像偏差为 0，但实际上每个都有偏差，而且都还不小！此时很直接的想法便是使用方差代替均值，这样就解决了正负抵消的问题，即使用

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

这时又引来了新的问题，对于 500 的均值来说，相差 5 其实是很小的(相差 1%），而对 20 的均值来说，5 相当于 25%的差异，这是使用方差也无法体现的。因此应该考虑改进上面的式子，让均值的大小不影响我们对差异程度的判断

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

上面这个式子已经相当好了。实际上这个式子就是开方检验使用的差值衡量公式。当提供了数个样本的观察值 x_1，x_2，……x_i ，……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。

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。

此时我们仍然需要明白对特征选择来说原假设是什么，因为计算出的开方值越大，说明对原假设的偏离越大，我们越倾向于认为原假设的反面情况是正确的。我们能不能把原假设定为“词 t 与类别 c 相关“？原则上说当然可以，这也是一个健全的民主主义社会赋予每个公民的权利(笑），但此时你会发现根本不知道此时的理论值该是多少！你会把自己绕进死胡同。所以我们一般都使用”词 t 与

类别 c 不相关“来做原假设。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(k 值可以根据自己的需要选，这也是一个健全的民主主义社会赋予每个公民的权利）。

好，原理有了，该来个例子说说到底怎么算了。

比如说现在有 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(任谁都看得出来两者很相关，但很遗憾，我们是智慧生物，计算机不是，它一点也看不出来，想让它认识到这一点，只能让它算算看）。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

用下面的表格更清晰：

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

如果有些特点你没看出来，那我说一说，首先，A+B+C+D=N(这，这不废话嘛）。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。

好，那么理论值是什么呢？以包含“篮球”且属于“体育”类别的文档数为例。如果原假设是成立的，即“篮球”和体育类文章没什么关联性，那么在所有的文章中，“篮球”这个词都应该是等概率出现，而不管文章是不是体育类的。这个概率具体是多少，我们并不知道，但他应该体现在观察结果中(就好比抛硬币的概率是二分之一，可以通过观察多次抛的结果来大致确定），因此我们可以说这个概率接近

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

(因为 A+B 是包含“篮球”的文章数，除以总文档数就是“篮球”出现的概率，当然，这里认为在一篇文章中出现即可，而不管出现了几次）而属于体育类的文章数为 A+C，在这些个文档中，应该有

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

篇包含“篮球”这个词(数量乘以概率嘛）。

但实际有多少呢？考考你(读者：切，当然是 A 啦，表格里写着嘛……）。

此时对这种情况的差值就得出了(套用式(1) 的公式），应该是

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

同样，我们还可以计算剩下三种情况的差值 D_{12}，D_{21}，D_{22}，聪明的读者一定能自己算出来(读者：切，明明是自己懒得写了……）。有了所有观察值的差值，就可以计算“篮球”与“体育”类文章的开方值

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

把 D_{11}，D_{12}，D_{21}，D_{22} 的值分别代入并化简，可以得到

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

词 t 与类别 c 的开方值更一般的形式可以写成

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

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。

实际上式(2）还可以进一步化简，注意如果给定了一个文档集合(例如我们的训练集）和一个类别，则 N，M，N-M(即 A+C 和 B+D）对同一类别文档中的所有词来说都是一样的，而我们只关心一堆词对某个类别的开方值的大小顺序，而并不关心具体的值，因此把它们从式(2）中去掉是完全可以的，故实际计算的时候我们都使用

$$\chi^2(t,\ c)=\frac{(AD-BC)^2}{(A+B)(C+D)} \qquad \text{式(3)}$$

好啦，并不高深对不对？

针对英文纯文本的实验结果表明：作为特征选择方法时，开方检验和信息增益的效果最佳(相同的分类算法，使用不同的特征选择算法来得到比较结果）；文档频率方法的性能同前两者大体相当，术语强度方法性能一般；互信息方法的性能最差(文献[17]）。

但开方检验也并非就十全十美了。回头想想 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。

好啦，关于开方检验先说这么多，有机会还将介绍其他的特征选择算法。

附：给精通统计学的同学多说几句，式(1) 实际上是对连续型的随机变量的差值计算公式，而我们这里统计的“文档数量“显然是离散的数值(全是整数），因此真正在统计学中计算的时候，是有修正过程的，但这种修正仍然是只影响具体的开方值，而不影响大小的顺序，故文本分类中不做这种修正。

(十一) 特征选择方法之信息增益

前文提到过，除了开方检验(CHI) 以外，信息增益(IG，Information Gain) 也是很有效的特征选择方法。但凡是特征选择，总是在将特征的重要程度量化之后再进行选择，而如何量化特征的重要性，就成了各种方法间最大的不同。开方检验中使用特征与类别间的关联性来进行这个量化，关联性越强，特征得分越高，该特征越应该被保留。

在信息增益中，重要性的衡量标准就是看特征能够为分类系统带来多少信息，带来的信息越多，该特征越重要。

因此先回忆一下信息论中有关信息量(就是"熵") 的定义。说有这么一个变量 X，它可能的取值有 n 多种，分别是 x_1，x_2，……，x_n，每一种取到的概率分别是 P_1，P_2，……，P_n，那么 X 的熵就定义为：

$$H(X) = -\sum_{i=1}^{n} P_i \bullet \log_2 P_i$$

意思就是一个变量可能的变化越多(反而跟变量具体的取值没有任何关系，只和值的种类多少以及发生概率有关) ，它携带的信息量就越大(因此我一直觉得我们的政策法规信息量非常大，因为它变化很多，基本朝令夕改，笑) 。

对分类系统来说，类别 C 是变量，它可能的取值是 C_1，C_2，……，C_n，而每一个类别出现的概率是 $P(C_1)$ ，$P(C_2)$ ，……，$P(C_n)$ ，因此 n 就是类别的总数。此时分类系统的熵就可以表示为：

$$H(C) = \sum_{i=1}^{n} P(C_i) \bullet \log_2 P(C_i)$$

有同学说不好理解呀，这样想就好了，文本分类系统的作用就是输出一个表示文本属于哪个类别的值，而这个值可能是 C_1，C_2，……，C_n，因此这个值所携带的信息量就是上式中的这么多。

信息增益是针对一个一个的特征而言的，就是看一个特征 t，系统有它和没它的时候信息量各是多少，两者的差值就是这个特征给系统带来的信息量，即增益。系统含有特征 t 的时候信息量很好计算，就是刚才的式子，它表示的是包含所有特征时系统的信息量。

问题是当系统不包含 t 时，信息量如何计算？我们换个角度想问题，把系统要做的事情想象成这样：说教室里有很多座位，学生们每次上课进来的时候可以随便坐，因而变化是很大的(无数种可能的座次情况) ；但是现在有一个座位，看黑板很清楚，听老师讲也很清楚，于是校长的小舅子的姐姐的女儿托关系(真辗转啊) ，把这个座位定下来了，每次只能给她坐，别人不行，此时情况怎样？对于座次的可能情况来说，我们很容易看出以下两种情况是等价的：(1) 教室里没有这个座位；(2) 教室里虽然有这个座位，但其他人不能坐(因为反正它也不能参与到变化中来，它是不变的) 。

对应到我们的系统中，就是下面的等价：(1) 系统不包含特征 t；(2) 系统虽然包含特征 t，但是 t 已经固定了，不能变化。

我们计算分类系统不包含特征 t 的时候，就使用情况(2) 来代替，就是计算当一个特征 t 不能变化时，系统的信息量是多少。这个信息量其实也有专门的名称，就叫做“条件熵”，条件嘛，自然就是指“t 已经固定“这个条件。

但是问题接踵而至，例如一个特征 X，它可能的取值有 n 多种(x_1，x_2，……，x_n) ，当计算条件熵而需要把它固定的时候，要把它固定在哪一个值上呢？答案是每一种可能都要固定一下，计算 n 个值，然后取均值才是条件熵。而取均值也不是简单的加一加然后除以 n，而是要用每个值出现的概率来算平均(简单理解，就是一个值出现的可能性比较大，固定在它上面时算出来的信息量占的比重就要多一些) 。

因此有这样两个条件熵的表达式：

$$H(C|X=x_i)$$

这是指特征 X 被固定为值 x_i 时的条件熵，

$$H(C|X)$$

这是指特征 X 被固定时的条件熵，注意与上式在意义上的区别。从刚才计算均值的讨论可以看出来，第二个式子与第一个式子的关系就是：

$$H(C|X)=P_1H(C|X=x_1)+P_2H(C|X=x_2)+\ldots+P_nH(C|X=x_n)$$

$$=\sum_{i=1}^{n}P_iH(C|X=x_i)$$

具体到我们文本分类系统中的特征 t，t 有几个可能的值呢？注意 t 是指一个固定的特征，比如他就是指关键词"经济"或者"体育"，当我们说特征"经济"可能的取值时，实际上只有两个，"经济"要么出现，要么不出现。一般的，t 的取值只有 t(代表 t 出现) 和 $\bar{t}$(代表 t 不出现)，注意系统包含 t 但 t 不出现与系统根本不包含 t 可是两回事。

因此固定 t 时系统的条件熵就有了，为了区别 t 出现时的符号与特征 t 本身的符号，我们用 T 代表特征，而用 t 代表 T 出现，那么：

$$H(C|T)=P(t)H(C|t)+P(\bar{t})H(C|\bar{t})$$

与刚才的式子对照一下，含义很清楚对吧，P(t) 就是 T 出现的概率，$P(\bar{t})$ 就是 T 不出现的概率。这个式子可以进一步展开，其中的

$$H(C|t)=-\sum_{i=1}^{n}P(C_i|t)\log_2 P(C_i|t)$$

另一半就可以展开为：

$$H(C|\bar{t})=-\sum_{i=1}^{n}P(C_i|\bar{t})\log_2 P(C_i|\bar{t})$$

因此特征 T 给系统带来的信息增益就可以写成系统原本的熵与固定特征 T 后的条件熵之差：

$$IG(T)=H(C)-H(C|T)$$
$$=-\sum_{i=1}^{n}P(C_i)\log_2 P(C_i)+$$
$$P(t)\sum_{i=1}^{n}P(C_i|t)\log_2 P(C_i|t)+P(\bar{t})\sum_{i=1}^{n}P(C_i|\bar{t})\log_2 P(C_i|\bar{t})$$

公式中的东西看上去很多，其实也都很好计算。比如 $P(C_i)$ ，表示类别 C_i 出现的概率，其实只要用 1 除以类别总数就得到了(这是说你平等的看待每个类别而忽略它们的大小时这样算，如果考虑了大小就要把大小的影响加进去) 。再比如 $P(t)$ ，就是特征 T 出现的概率，只要用出现过 T 的文档数除以总文档数就可以了，再比如 $P(C_i|t)$ 表示出现 T 的时候，类别 C_i 出现的概率，只要用出现了 T 并且属于类别 C_i 的文档数除以出现了 T 的文档数就可以了。

从以上讨论中可以看出，信息增益也是考虑了特征出现和不出现两种情况，与开方检验一样，是比较全面的，因而效果不错。但信息增益最大的问题还在于它只能考察特征对整个系统的贡献，而不能具体到某个类别上，这就使得它只适合用来做所谓"全局"的特征选择(指所有的类都使用相同的特征集合) ，而无法做"本地"的特征选择(每个类别有自己的特征集合，因为有的词，对这个类别很有区分度，对另一个类别则无足轻重) 。

看看，导出的过程其实很简单，没有什么神秘的对不对。可有的学术论文里就喜欢把这种本来很直白的东西写得很晦涩，仿佛只有读者看不懂才是作者的真正成功。

咱们是新一代的学者，咱们没有知识不怕被别人看出来，咱们有知识也不怕教给别人。所以咱都把事情说简单点，说明白点，大家好，才是真的好。

(十二) 特征选择与特征权重计算的区别

在文本分类的过程中，特征(也可以简单的理解为“词”) 从人类能够理解的形式转换为计算机能够理解的形式时，实际上经过了两步骤的量化——特征选择阶段的重要程度量化和将具体文本转化为向量时的特征权重量化。初次接触文本分类的人很容易混淆这两个步骤使用的方法和各自的目的，因而我经常听到读者有类似“如何使用 TFIDF 做特征选择”或者“卡方检验量化权重后每篇文章都一样”等等困惑。

文本分类本质上也是一个模式识别的问题，因此我想借用一个更直观的例子来说说特征选择和权重量化到底各自是什么东西，当然，一旦解释清楚，你马上就会觉得文本分类这东西实在白痴，实在没什么技术含量，你也就不会再继续看我的技术博客，不过我不担心，因为你已经踏上了更光明的道路(笑) ，我高兴还来不及。

想想通过指纹来识别一个人的身份，只看一个人的指纹，当然说不出他姓甚名谁，识别的过程实际上是比对的过程，要与已有的指纹库比较，找出相同的，或者说相似到一定程度的那一个。

首要的问题是，人的指纹太复杂，包含太多的位置和几何形状，要完全重现一个人的指纹，存储和计算都是大麻烦。因此第一步总是一个特征选择的问题，我们把全人类的指纹都统计一下，看看哪几个位置能够最好的区分不同的人。显然不同的位置效果很不一样，在有的位置上，我的指纹是是什么形状，其他人也大都是这个形状，这个位置就不具有区分度，或者说不具有表征性，或者说，对分类问题来说，它的重要程度低。这样的位置我们就倾向于在识别的时候根本不看它，不考虑它。

那怎么看谁重要谁不重要呢？这就依赖于具体的选择方法如何来量化重要程度，对卡方检验和信息增益这类方法来说，量化以后的得分越大的特征就越重要(也就是说，有可能有些方法，是得分越小的越重要) 。

比如说你看 10 个位置，他们的重要程度分别是：

1 2 3 4 5 6 7 8 9 10

(20，5，10，20，30，15，4，3，7， 3)

显然第 1，第 3，4，5，6 个位置比其他位置更重要，而相对的，第 1 个位置又比第 3 个位置更重要。

识别时，我们只在那些重要的位置上采样。当今的指纹识别系统，大都只用到人指纹的 5 个位置(惊讶么？只要 5 个位置的信息就可以区分 60 亿人），这 5 个位置就是经过特征选择过程而得以保留的系统特征集合。假设这个就是刚才的例子，那么该集合应该是：

(第 1 个位置，第 3 个位置，第 4 个位置，第 5 个位置，第 6 个位置)

当然，具体的第 3 个位置是指纹中的哪个位置你自己总得清楚。

确定了这 5 个位置之后，就可以把一个人的指纹映射到这个只有 5 个维度的空间中，我们就把他在 5 个位置上的几何形状分别转换成一个具体的值，这就是特征权重的计算。依据什么来转换，就是你选择的特征权重量化方法，在文本分类中，最常用的就是 TFIDF。

我想一定是“权重“这个词误导了所有人，让大家以为 TFIDF 计算出的值代表的是特征的重要程度，其实完全不是。例如我们有一位男同学，他的指纹向量是：

(10，3，4，20，5)

你注意到他第 1 个位置的得分(10) 比第 3 个位置的得分(3) 高，那么能说第 1 个位置比第 3 个位置重要么？如果再有一位女同学，她的指纹向量是：

(10，20，4，20，5)

看看，第 1 个位置得分(10) 又比第 3 个位置(20) 低了，那这两个位置到底哪个更重要呢？答案是第 1 个位置更重要，但这不是在特征权重计算这一步体现出来的，而是在我们特征选择的时候就确定了，第 1 个位置比第 3 个位置更重要。

因此要记住，通过 TFIDF 计算一个特征的权重时，该权重体现出的根本不是特征的重要程度！

那它代表什么？再看看两位同学的指纹，放到一起：

(10， 3，4，20，5)

(10，20，4，20，5)

在第三个位置上女同学的权重高于男同学，这不代表该女同学在指纹的这个位置上更“优秀“(毕竟，指纹还有什么优秀不优秀的分别么，笑），也不代表她的这个位置比男同学的这个位置更重要，3 和 20 这两个得分，仅仅代表他们的”不同“。

在文本分类中也是如此，比如我们的系统特征集合只有两个词：

(经济，发展)

这两个词是使用卡方检验(特征选择）选出来的，有一篇文章的向量形式是

(2，5)

另一篇

(3，4)

这两个向量形式就是用 TFIDF 算出来的，很容易看出两篇文章不是同一篇，为什么？因为他们的特征权重根本不一样，所以说权重代表的是差别，而不是优劣。想想你说“经济这个词在第二篇文章中得分高，因此它在第二篇文章中比在第一篇文章中更重要“，这句话代表什么意义呢？你自己都不知道吧(笑）。

所以，当再说起使用 TFIDF 来计算特征权重时，最好把“权重“这个字眼忘掉，我们就把它说成计算得分好了(甚至”得分“也不太好，因为人总会不自觉的认为，得分高的就更重要），或者就仅仅说成是量化。

如此，你就再也不会拿 TFIDF 去做特征选择了。

小 Tips：为什么有的论文里确实使用了 TFIDF 作特征选择呢？

严格说来并不是不可以，而且严格说来只要有一种方法能够从一堆特征中挑出少数的一些，它就可以叫做一种特征选择方法，就连“随机选取一部分“都算是一种，而且效果并没有差到惊人的地步哦！还是可以分对一大半的哦！所以有的人就用 TFIDF 的得分来把特征排排序，取得分最大的几个进入系统特征集合，效果也还行(毕竟，连随机选取效果也都还行），怎么说呢，他们愿意这么干就这么干吧。就像咱国家非得实行户口制度，这个制度说不出任何道理，也不见他带来任何好处，但不也没影响二十一世纪成为中国的世纪么，呵呵。

关于复旦大学自然语言处理实验室的基准语料

使用复旦大学基准语料库所做的对比实验并非我本人进行的，我只是引用了文献“周文霞：现代文本分类技术研究，武警学院学报，2007.12”的实验结果。因此我手头没有该文作者所使用的预处理程序。但复旦大学的语料库在中科院中文自然语言处理开放平台上有提供下载，页面地址是http://www.nlp.org.cn/docs/doclist.php?cat_id=16&type=15，可能需要注册用户，待管理员审批完成之后方可下载。我已经下载了一份，训练集与测试集共 100MB的样子，大家有需要的话也可以想办法分发给大家。

另外,搜狗实验室提供的文本分类语料库也有在线下载版本,地址是http://www.sogou.com/labs/dl/c.html,共有 10 个类别,8 万篇左右的文本.

在此只是提醒大家，文本分类语料库的建立是需要很多人力成本的，无论复旦大学还是搜狗实验室，既然免费与大家共享，就希望大家在使用的时候至少注明出处，也不枉别人对我们的信任。

谢谢。

P.S.实在没有办法下载到的朋友也可以加我的QQ49900829，在消息中注明需要复旦语料库，我可以在线传送给大家。

复旦大学语料库的一些统计信息

复旦大学的中文语料库分为训练集和验证集两部分,两部分的文档数量基本相等,但现在做测评一般都不采用这种预先划分的方法,而多用交叉验证,因此在将训练集与验证集合并之后,得到该语料库的一些基本信息如下:

类别总数量:20

文档总数量:19637

类别名称(类别代码) :文档数量

Agriculture(C32) :2043 篇

Art(C3) :1482 篇

Communication(C17) :52 篇

Computer(C19) :2715 篇

Economy(C34) :3201 篇

Education(C5) :120 篇

Electronics(C16) :55 篇

Energy(C15) :65 篇

Enviornment(C31) :2435 篇

History(C7) :934 篇

Law(C35) :103 篇

Literature(C4) :67 篇

Medical(C36) :104 篇

Military(C37) :150 篇

Mine(C23) :67 篇

Philosophy(C6) :89 篇

Politics(C38) :2050 篇

Space(C11) :1282 篇

Sports(C39) :2507 篇

Transport(C29) :116 篇

同时,在使用 ictclas4j 分词包对其进行分词的过程中,发现复旦语料库中存在一些文章会使得 ictclas4j 报错,其中因为分词包本身字库缺少某些文字,以及一些神秘的字符组合(确实很神秘) 会导致分词过程出错,因此能够被成功分词而供后续使用的文档数并不如上面所列这么多,在分词之后,情况如下:

类别总数量:20

文档总数量:18185

类别名称(类别代码) :文档数量

Agriculture(C32) :1949 篇

Art(C3) :1237 篇

Communication(C17) :52 篇

Computer(C19) :2591 篇

Economy(C34) :2912 篇

Education(C5) :111 篇

Electronics(C16) :51 篇

Energy(C15) :63 篇

Environment(C31) :2347 篇

History(C7) :708 篇

Law(C35) :103 篇

Literature(C4) :65 篇

Medical(C36) :98 篇

Military(C37) :147 篇

Mine(C23) :63 篇

Philosophy(C6) :86 篇

Politics(C38) :1920 篇

Space(C11) :1226 篇

Sports(C39) :2344 篇

Transport(C29) :112 篇

在已分词后的语料库里,可以看出这样几个特点,一,文档总数比未分词的版本少了 1448 篇(可见 ictclas4j 的错误还是满普遍的) ；第二，文档数量分布仍不均衡，最多的经济类文章有 2912 篇，而最少的电子类与通信类文章仅有 51 篇与

52 篇，往好的方向说可以考察你所开发的系统如何应对数据集偏斜的问题，往坏的方向说给要上线的系统作训练集恐怕不太合适。

在下一篇文章中, 我将进一步总结词频统计的结果.

复旦大学语料库的一些统计信息Part 2 词频

经过词频统计,看到复旦大学中文语料库的总词数为 116558 个(而且还是去掉了停止词及代词,介词,数词和时间短语等无关内容之后的结果),数量十分巨大.而各个类别的词汇数量分别为:

类别名称：Agriculture 总文档数：1949 总词数：29163

类别名称：Art 总文档数：1237 总词数：40816

类别名称：Communication 总文档数：52 总词数：2283

类别名称：Computer 总文档数：2591 总词数：19340

类别名称：Economy 总文档数：2912 总词数：37021

类别名称：Education 总文档数：111 总词数：5719

类别名称：Electronics 总文档数：51 总词数：2693

类别名称：Energy 总文档数：63 总词数：2848

类别名称：Environment 总文档数：2347 总词数：25155

类别名称：History 总文档数：708 总词数：47205

类别名称：Law 总文档数：103 总词数：3834

类别名称：Literature 总文档数：65 总词数：5844

类别名称：Medical 总文档数：98 总词数：3877

类别名称：Military 总文档数：147 总词数：4615

类别名称：Mine 总文档数：63 总词数：3708

类别名称：Philosophy 总文档数：86 总词数：5190

类别名称：Politics 总文档数：1920 总词数：35292

类别名称：Space 总文档数：1226 总词数：14557

类别名称：Sports 总文档数：2344 总词数：42665

类别名称：Transport 总文档数：112 总词数：4644

很容易看出词汇的数量基本与类别包含的文档数成正比,但也有一些极其特殊的类别,比如艺术(Art) 和历史(History) ,其文档数量仅有计算机文章数量的一半,但包含的词汇量却是计算机类别的两倍以上(分别是 40816:19340 和 47205:19340,尤以历史类文章为甚,其文档数量仅有计算机类的三分之一还不到) 。

直观上的想法是，历史和艺术类文章包含了大量的人名，地名或者事件名等专有名词，因而词汇数量上表现得很巨大。计算机类文章包含词汇较少，一是因为其为新兴学科，包含的内容本就较少，另一个更重要的原因则在于前期对文章的处理忽略了所有的英文单词及缩写，而这些内容在计算机相关的文章中所占比重很大。

如果我们看整个语料库出现次数最多的十个词，会发现他们大致也是我们的国计民生所关注的几个方面(巧合？未必！）它们是：

词内容：经济 词性：名词 词频：233906 文档频率：8975
词内容：发展 词性：动词 词频：189181 文档频率：11847
词内容：农业 词性：名词 词频：126603 文档频率：4105
词内容：社会 词性：名词 词频：108988 文档频率：8686
词内容：政治 词性：名词 词频：106847 文档频率：4971
词内容：大 词性：形容词 词频：106111 文档频率：14729
词内容：中国 词性：名词 词频：105269 文档频率：10885
词内容：人 词性：名词 词频：98034 文档频率：11037
词内容：问题 词性：名词 词频：94458 文档频率：12538
词内容：个 词性：量词 词频：91717 文档频率：14428

通过与某些类别中排名前十位的词对比，我们可以看出很多问题，例如计算机类别：

词内容：系统 词性：形容词 词频：45496 文档频率：2244
词内容：控制 词性：动词 词频：21937 文档频率：1734
词内容：图 词性：名词 词频：20396 文档频率：1914
词内容：方法 词性：名词 词频：20073 文档频率：2141
词内容：个 词性：量词 词频：19661 文档频率：2207
词内容：算法 词性：名词 词频：18879 文档频率：1336
词内容：数据 词性：名词 词频：17691 文档频率：1357
词内容：模型 词性：名词 词频：17182 文档频率：1423

词内容：网络 词性：名词 词频：16980 文档频率：1159

词内容：进行 词性：动词 词频：16406 文档频率：2094

词内容：问题 词性：名词 词频：14617 文档频率：1965

再比如交通类别：

词内容：铁路 词性：名词 词频：280 文档频率：51

词内容：运输 词性：动词 词频：205 文档频率：74

词内容：交通 词性：名词 词频：158 文档频率：54

词内容：大 词性：形容词 词频：147 文档频率：59

词内容：工程 词性：名词 词频：136 文档频率：31

词内容：个 词性：量词 词频：117 文档频率：51

词内容：年 词性：量词 词频：114 文档频率：52

词内容：建设 词性：动词 词频：108 文档频率：40

词内容：公路 词性：名词 词频：106 文档频率：34

词内容：条 词性：量词 词频：105 文档频率：38

我们会发现，

第一：整个语料库出现最多的词未必在各个类别中也最多，实际上通过计算机和交通类别可以看出，几乎完全不同！这意味着在进行文本分类的训练阶段，针对各个类取不同的特征集合(即所谓 local 的特征选择) 很有必要，如果所有的类别都使用相同的特征集合(而且毫无悬念的，这个特征集合就是语料库的特征集合），那么分类效果会因为没有为各个类找到最佳的特征而大打折扣；

第二，注意到“个”这个词出现在所有类别排名靠前的词汇中间，但直觉告诉我们，这个词很难对分类产生什么贡献(行话叫区分度很差）。此结论与信息论中所说的“一个词分布越广越均匀，则区分度越差”是一个意思。当然，在这里“个”会如明星般的出现在所有类别靠前的位置上，完全是因为我们的排名是根据词频来统计的(根据文档频率排序也会产生相似的结果），而使用像开方检验，信息增益这样的特征选择算法，就是为了避免这种区分度差的词出现在最终的特征集合中，从而影响分类效果。

在后续的文章里，我还会给出使用了开方检验计算特征得分以后的排名情况，“个”这个词会不会从前十名中消失呢？又有哪些词会冲上头排呢？我们拭目以待。（音乐响，幕布缓慢拉上，灯光渐暗）

复旦大学语料库的一些统计信息Part 3 文档频率预处理

词的文档频率(DF,即一个词在多少篇文档中出现) 虽然并不用于真正的特征选择,但是作为特征选择前的预处理手段还是经常被使用,因为出现次数太少的词(低频词,或者叫生僻词) 往往是表意能力很差的词,更极端的情况下,那种在几万篇文档中却只出现几次的词更有可能是作者的笔误(即创造了一个不存在的词) ,使用它的更大好处还在于可以大大消减文档集中需要处理的词汇数量.请看以下的数据,在上一篇文章中对复旦语料库进行分词,去停止词,去无用词性的词的基础上,再进行一次根据 DF 的处理,去除所有文档频率小于等于 3 的词,得到的对比结果如下.

文档频率筛选前	文档频率筛选后
总词数 116558 类别名称：Agriculture 总词数：29163 类别名称：Art 总词数：40816 类别名称：Communication 总词数：2283	总词数 50283 类别名称：Agriculture 总词数：23258 类别名称：Art 总词数：30899 类别名称：Communication 总词数：2207

类别名称：Computer 总词数：19340	类别名称：Computer 总词数：15545
类别名称：Economy 总词数：37021	类别名称：Economy 总词数：28363
类别名称：Education 总词数：5719	类别名称：Education 总词数：5437
类别名称：Electronics 总词数：2693	类别名称：Electronics 总词数：2604
类别名称：Energy 总词数：2848	类别名称：Energy 总词数：2702
类别名称：Environment	类别名称：Environment

总词数：25155 类别名称：History 总词数：47205 类别名称：Law 总词数：3834 类别名称：Literature 总词数：5844 类别名称：Medical 总词数：3877 类别名称：Military 总词数：4615	总词数：19781 类别名称：History 总词数：31436 类别名称：Law 总词数：3656 类别名称：Literature 总词数：5500 类别名称：Medical 总词数：3566 类别名称：Military 总词数：4256

类别名称：Mine	类别名称：Mine
总词数：3708	总词数：3507
类别名称：Philosophy	类别名称：Philosophy
总词数：5190	总词数：4968
类别名称：Politics	类别名称：Politics
总词数：35292	总词数：26046
类别名称：Space	类别名称：Space
总词数：14557	总词数：12136
类别名称：Sports	类别名称：Sports
总词数：42665	总词数：30803
类别名称：Transport	类别名称：Transport
总词数：4644	总词数：4276

怎么样?总词数从 116558 下降到 50283,减少了一多半.可见生僻词还是广泛存在的.而具体到各个类别上也各不相同.减少的比较少的典型类别例如法律类,仅仅消减掉了 4.6%的词汇,而历史类整整去除了 33.4%!这倒也不难想象,因为法律概念一般有标准的名称和说法,文章也都有通用的格式或成文的套路遵循,作者本身发挥的余地不大.而历史类文章包含大量的人名地名和事件名称,这些名称出现数量多但每一个出现的次数相对较少,而同时人名地名也很难作为区分文章主题的依据(出现"诸葛"就一定是说三国时期的事情么?我们计算所有位老师就姓诸葛,还恰好是搞自然语言处理方面的,呵呵) ,筛掉它们我们也不心疼.

有了这一步处理,又可以为开方检验的计算增添不少方便.

(音乐再次响起，幕布再次缓慢拉上，灯光渐暗)

复旦大学语料库的一些统计信息Part4 开方检验

使用开方检验能够修正文档频率作为特征选择手段的一些不足,在对复旦大学语料库作过一系列处理之后,为 20 个类别分别计算各自特征的开方值并排序(开方值越大则说明越应该作为特征被选中) 之后,可以看出很多有意思的东西.记得在这一系列文章的 part2 中提到过仅仅使用词频来排序的时候,”个”这个词如明星般的在很多类别中都频繁出现在排名前十的位置上,但这个词实际上没有表意功能,对分类贡献不大,是理应被特征选择程序筛选掉的.使用开方检验方法后,我们惊喜的发现(读者:切!前人早都发现无数次了……) ：“个”消失了!

我稍微摘选结果中的几个类别在词频排序和开方值排序之间的比较,大家一起来瞅瞅。(前面也说过了，使用词频排序和使用文档频率情况大体相同，因此不再单独列出)

历史类别(History)

词频排序	开方值排序
历史 词频：24303	近代史词频：350
中国 词频：15146	史学 词频：2566
人 词频：11707	现代史词频：164
社会 词频：8655	史料 词频：529
发展 词频：8540	历史学词频：771
研究 词频：8007	世界史词频：169
文化 词频：7607	史实 词频：294
大 词频：6748	战争 词频：2095

新　词频：6706	封建　词频：1156
到　词频：6537	历史学词频：386
说　词频：6462	人物　词频：2399
种　词频：5694	统治　词频：1056
问题 词频：5304	侵略　词频：501
政治 词频：5178	记载　词频：625
文学 词频：5176	历史　词频：24303
年　词频：4830	斗争　词频：1731
经济 词频：4810	帝国主义词频：655
思想 词频：4550	清政府词频：289
这种 词频：4476	王朝　词频：370
个　词频：4276	民族　词频：4168

我列出了历史类文章中两种方法排名前二十的词汇，可以发现使用词频(或者文档频率）统计的结果纯粹无聊(简直无聊，特别无聊），除了“历史”，“社会”，“发展”听着还像那么回事以外，什么“说”，“种”，“年”这样的词真该统统杀光光。

用了开方检验就果然不一样，看看“史料”啊，“记载”呀，“王朝”呀，多正儿八经的历史词汇！我真是太喜欢开方检验啦！(笑)

当然结果也未必就十全十美了，我举个计算机的例子给你看。

计算机类别(Computer)

词频排序	开方值排序
系统词频：45496	算法　词频：18879
控制词频：21937	自动化词频：2674
图　词频：20396	计算机词频：7569
方法词频：20073	函数　词频：9932
个　词频：19661	定义　词频：9817
算法词频：18879	关键词词频：1956
数据词频：17691	软件　词频：6189
模型词频：17182	引言　词频：937
网络词频：16980	集合　词频：3717
进行词频：16406	输入　词频：6385
问题词频：14617	摘　词频：1540
应用词频：13883	定理　词频：4487
对象词频：13656	模型　词频：17182
信息词频：13468	用户　词频：10053
结构词频：12658	参数　词频：8491
研究词频：12308	导师　词频：969
实现词频：11331	向量　词频：2658

过程词频：11293	期　词频：213
设计词频：10713	输出　词频：6149
种　词频：10506	矩阵　词频：5431

看见”摘”这个词了么?居然出现在第 11 位,现在我还要告诉你,如果不是在去停止词的阶段把”要”字给去掉了,”要”字也会出现在”摘”附近的位置上,聪明的读者应该能大致猜出几分原因了吧.没错,到复旦语料库的计算机类文档中稍稍察看就会发现,大量的文档都有类似这样的格式:

计算机应用

COMPUTER APPLICATIONS

1999 年第 19 卷第 6 期 Vol.19 No.6 1999

一种基于智能 Agent 的协同工作模型

朱晓芸　何钦铭　王申康

摘　要　计算机支持的协同工作(CSCW) 需要研究出适应各种协同工作方式的灵活、开放、可扩充的模型结构。本文以分布式人工智能研究中的智能 Agent 为系统基本单元，提出一种基于智能 Agent 的协同工作模型，给出了它的具体实现。

关键词　计算机支持的协同工作，智能 Agent，分布式人工智能

AN INTELLIGENT AGENT BASED COLLABORATIVE WORK MODEL

Zhu XiaoyunHe QinmingWang Shenkang

看到”摘要”的位置了么?一来复旦语料库计算机类的文档大都是这类期刊文献的形式,因此”摘要”这个词频繁出现;二来其他类别的语料虽然也都有大量以文献

作为来源的文档,但甚少用到”摘要”这个词;最后一点,注意到原文中”摘要”两个字是被空格分开的,只有我们这些地球的主宰者,全能的人类才能看出他们是一个词,而我们使用的分词程序会毫不留情的将其判断为”摘”和”要”两个词.这三点综合作用的结果,就使得我们的程序认为”摘”这个词对计算机类文档有很强的代表性(当然，我们自己心里知道，这纯属无稽之谈）,从而入选了特征的 TOP20。

以上分析给我们的启示是：作为训练集的文档来源一定要广泛，如果计算机类的文章还包括教科书，网页，个人博客的内容，显然就不会出现“摘”字这种笑话；另一方面，再一次重申，文本分类就应该是只依据文本的内容，而不应该包含文件的编码，文章格式，发表时间等外部信息，“摘”字的笑话多少也是因为文章的格式(在“摘”和“要”之间总有空格）影响了分词程序的判断而致。

关于复旦语料库所说的这些东西有点杂，有机会的话我会重新整理，再结合特征选择的具体方法，把特征选择的过程说说清楚。

以上。

10 分钟开始使用ICTCLAS Java版

ICTCLAS 是中科院计算所出品的中文分词程序包，在国内一直有着良好的口碑和很高的使用率。之前一直只有 C++的版本提供，而现在 C#，Delphi 和 Java 版本已经纷纷出炉。下面用一个极小的例子，让大家 10 分钟之内就能用上 ICTCLAS ，从此也开始自己的文本分类和搜索引擎开发之路。

需要首先说明的是，不同于以前的 C++版提供的 JNI 调用，本次使用的是纯 Java 版本的 ICTCLAS，下载地址在 http://ictclas.org/Down_OpenSrc.asp。

好，假设你已经下载了我们需要使用的 Java 版本 ictclas4j，现在把它解压缩，然后把 Data 文件夹整个拷贝到 Eclipse 项目的文件夹下，而 bin 目录下的 org 文件夹整个拷贝到你 Eclipse 项目的 bin 目录下，把 src 目录下的 org 文件夹整个拷贝到 Eclipse 项目的 src 目录下(最简单快捷的使用方式，或者你自己打成 jar 包，这样无论放到哪里，都可以在 build path 里面导入这个 jar 包啦) 。

现在就可以在你的项目里新建一个类来试试。我新建了一个类，代码如下：

```
import org.ictclas4j.bean.SegResult;

import org.ictclas4j.segment.SegTag;

public class OneMain {

public static void main(String[] args)  {

System.out.println("This is OneMain") ;

SegTag st = new SegTag(1) ;

SegResult sr = st
```

```
.split("一块勤奋地漂亮的一块钱,/打造经济的航空母舰。ABCD.#$% Hello World!\n 又一段文本 123 辆 ！ 3.0") ;

System.out.println(sr.getFinalResult() ) ;

}

}
```

很显然文本“一块勤奋地漂亮的一块钱,/打造经济的航空母舰。ABCD.#$% Hello World!"n 又一段文本 123 辆 ！ 3.0”就是我们用来测试的文本，其中包含了中文，英文，标点符号，乱七八糟符号(笑) 及阿拉伯数字。

我们运行刚才的程序，看下输出结果：

```
This is OneMain

一块/s 勤奋/a 地/u 漂亮/a 的/u 一/m 块/q 钱/n ,/w //nx 打造/v 经济/n 的/u 航空母舰/n 。/w ABCD.#$%/nx Hello/nx World/nx !/w 又/d 一/m 段/q 文本/n 123/m 辆/q
```

看到了么，分词的结果是一个长长的 String 类数据，用空格区分出每个词，每个词还用/后面的英文标号标出了词性。一起来看看几个有趣的地方。

原文中其实有两个“一块”，一处是“一块勤奋”，这里很正确的识别为了副词，而后面的“一块钱”中的“一块”也正确的识别为数量词。

阿拉伯数字正确识别为数词，包括小数形式的“3.0”。而英文和乱七八糟符号(包括那个不可见的换行符，你找到它在哪了吗？) 则都被划为一类——/nx!

(因为我也不知道 ICTCLAS 内部人员管它叫什么啦，非法字符啊，还是无效字符啊，或者其它字符啊，名字可以自己取嘛)

测试文本中还有两个叹号，一个是英文半角的!，一个是中文全角的！，两者也都被正确识别为标点符号，但英文的句号“.“就被认为是/nx 啦。

测试文本中的空格被完全忽略。

好，十分简单对不对？去玩玩吧。

参考文献

文本分类入门系列所有文章的参考文献集中列于此,其他文章中再做引用时仅列出标号,不再重复写出作者和出版物,出版年份等信息.

[1]李晓明，闫宏飞，王继民，“搜索引擎——原理、技术与系统”.科学出版社，2004

[2]冯是聪, ″中文网页自动分类技术研究及其在搜索引擎中的应用,″ 北京大学, 博士论文, 2003

[3]Y. Yang and X. Liu, ″A re-examination of text categorization methods″ presented at Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99) , 1999.

[4]F. Sebastiani, ″A tutorial on Automated Text Categorization″, Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence, Buenos Aires, AR, 1999

[5]王涛：文本自动分类研究，图书馆学研究，2007.12

[6]周文霞：现代文本分类技术研究，武警学院学报，2007.12

[7]奉国和：自动文本分类技术研究，情报杂志，2007.12

[8]崔彩霞，张朝霞：文本分类方法对比研究，太原师范学院学报(自然科学版），2007.12

[9]吴军：Google 黑板报数学之美系列，http://googlechinablog.com

[10]刘霞，卢苇：SVM 在文本分类中的应用研究，计算机教育，2007.1

[11]都云琪，肖诗斌：基于支持向量机的中文文本自动分类研究,计算机工程，2002，28(11)

[12]周昭涛，卜东波：文本的图表示初探，中文信息学报，第 19 卷 第 2 期

[13]Baeza-Yates,R.and Ribeiro-Neto:Modern Information Retrieval,1st ed.Addison Wesley Longman,Reading,MA,1999

[14]唐春生，张磊：文本分类研究进展

[15]李蕴，罗振声：基于语义相关和概念相关的自动分类方法研究，计算机工程与应用，2003.12

[16]单松巍，冯是聪，李晓明：几种典型特征选取方法在中文网页分类上的效果比较，计算机工程与应用，2003.22

[17]Yiming Yang,Jan O Pedersen:A comparative Study on Feature Selection in Text Categorization, Proceedings of the Fourteenth International Conference on Machine Learning(ICML~97) , 1997

[18]董振东：知网简介，知网，http://www.keenage.com/zhiwang/c_zhiwang.html

[19]Tom M.Mitchell,”Machine Learning”,McGraw Hill Companies，1997

[20] Edda Leopold, Jorg Kindermann,“Text Categorization with Support Vector Machines：How to Represent Texts in Input Space?”, Kluwer Academic Publishers,2002

[21] Thorsten Joachims,”Text Categorization with Support Vector Machines: Learning with Many Relevant Features”

[22]Nello Cristianini,An Introduction to Support Vector Machines and Other Kernel-based Learning Methods,Cambridge University Press,2000

[23]F. Sebastiani, ″MACHINE LEARNING IN AUTOMATED TEXT CATEGORIZATION″, ACM Computing Surveys, Vol. 34, No. 1, 2002

[24]TRS 公司，TRS 文本挖掘基础件白皮书

[25]苏金树，张博锋：基于机器学习的文本分类技术研究进展，Journal of Software,2006.9