北京交通大学

本科毕业设计（论文）

基于人工策略及机器学习算法的网页分类器模型的设计与过滤系统的实现

Design Of webpage classification model based on artificial strategy and machine learning and Implementation of filter system

学　　院：软件学院

专　　业：软件工程

学生姓名：XXX

学　　号：XXXXXXXX

指导教师：XXX

北京交通大学

2016 年 5 月

学士论文版权使用授权书

本学士论文作者完全了解北京交通大学有关保留、使用学士论文的规定。特授权北京交通大学可以将学士论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：　　指导教师签名：

签字日期：　　年　　月　　日　　签字日期：　　年　　月　　日

中文摘要

摘要：网页分类，是将网页按照既定的类型（如索引页，内容页，视频页等）进行分类，由于网页存在许多不同于传统文本分类的其他特征，如 HTML 的网页结构和网页中超文本链接的相关特征，使得网页分类非常不同于传统的文本分类，而对文本的内容进行分类是传统的文本分类的主要特点[1]，其主要采用切词及对词频的计算等特征进行文章内容的分类。本文首先通过对已有机器学习算法的比较和调研并利用网页分类特有的特征提供网页分类器模型的理论基础。 现如今互联网中网页繁多，网页数量已达千亿级，需要大批量地对网页进行识别并准确分类，须确保其时间复杂度及分类的准确率。本文在已有的理论基础之上以及通过实际项目的研发，提供一个综合人工策略模型和机器学习模型的网页分类器模型的设计和构建方法。为了让网页分类器模型不仅能够起到识别并分类离线网页的作用，还能自动甄别指定类型的线上网页并将该网页阻挡在网页库外，我们还为此设计了一个过滤系统，通过利用已开发的网页分类器模型进行实际的上线运行工作，本文的最后将会对这一过滤系统的实现进行概要性的描述。

关键词：网页分类； 机器学习； 人工策略； 过滤系统

ABSTRACT

ABSTRACT: Webpage classification, is to classify the webpages into different categories, the features in webpage classification is quite different from the features in traditional text classification, such as the structure of the HTML and the features of hyper-links. The traditional text classification is to classify mostly depending on the content of the text, and mainly using features like words segmentation and word frequency. In this paper, we first compare and have researches on different machine learning algorithms in webpage classification. Nowadays, there exist numerous webpages in the internet, when needs to classify these huge amount of webpages, it should guarantee the time complexity and accuracy in classifying. In our paper, we provide a design and construction methods of a model of webpage classification which consists of artificial strategy model and machine learning model on our real projects. In order not only to classify off-line webpages by using our models, but also to automatically identify webpages of specific category, and to stop them from getting into database of webpages, we designed a filter system, in the end of this paper, we will describe the design of the system synoptically.

KEYWORDS： webpage classification; machine learning; artificial strategy; filter system

目　　录

1 引言

1.1 课题背景

随着国内计算机互联网和信息技术的飞速发展，互联网上的网页数量急剧增长，1月22日上午发布第37次《中国互联网络发展状况统计报告》显示[12]，去年中国网页数量首次突破2000亿。本人在公司实习期间对公司离线网页库的调研中可知在打压低质网页相关项目启动前，网页库中已收录近1400亿网页。如此巨大的网页数量，对任一项网络信息检索相关技术都将是重大的挑战。

本论文题目来源于本人在公司实习的《网页库有效性优化》相关项目,该项目主要通过设计开发网页分类器模型(其中包括人工策略模型和机器学习模型)用于识别各种低质垃圾网页(如作弊页,空页面,报错页等),并将垃圾网页从网页库中删去,以提升网页库的有效性,降低垃圾占有率,并为节省大量存储空间。

本文将通过对不同机器学习算法的比较调研和已完成的对网页分类器模型的项目开发，提供一个网页分类器模型的设计及理论基础。该类网页分类器模型利用网页类型识别技术将网页进行识别并分类，已分好类的网页便可进一步得到有效利用，例如之前提及的本人所参与项目中的应用，即对低质垃圾网页的识别并准确地将垃圾网页删除和过滤，或是对索引页类型的识别，将已识别出的索引页用于搜索引擎抓取环节中的挖掘新链接等。

1.2 问题定义

网页类型识别技术，又称网页分类技术，是网络信息检索技术研究中的关键技术之一。能否从不断迅速膨胀的网页中快速、准确地搜索到用户感兴趣或用户真正想要的信息对网页分类技术同样是个巨大的挑战[3]。

分类是数据挖掘的一种非常重要的方法，分类器即是在已有数据的基础上学会一个分类函数或构造出一个分类模型，也叫做分类器[4]。网页分类器模型便是将原始网页通过其各种特征划分至已有设定的多种网页类型中。在原始的人工策略上，网页分类器模型依靠人为的通过网页特征指定相关规则编写网页分类器程序对网页进行筛选、识别、分类，而由于近年机器学习算法的不断快速发展，及越来越普遍的运用，网页分类同样可以使用机器学习算法，并极大的提高了网页分类的准确率。常见的分类算法包括决策树、逻辑回归、最大熵、朴素贝叶斯、神经网络等机器学习算法[13]。

1.3 研究现状与技术难点

由于网页数量的极速膨胀，如今的网页分类规模已不再同数年前搜索引擎开发前期那般，对网页分类的准确率要求也越来越高。通过原始的人为制定规则筛选识别网页类型的基于人工策略模型的网页分类器模型的准确率（大约在 50%~80%之间）已不能满足当今搜索引擎的需要。由于机器学习的不断应用于工程实践，网页分类器模型中也越来越多的使用了机器学习模型致使其准确率大幅度提升至大部分网页类型识别的准确率在 97%以上甚至更高。然而，机器学习模型的局限在于由于特征维度的提升和机器学习算法本身的高时间复杂度使得单纯的基于机器学习算法的网页分类器模型的运行时间太长同样不能满足如今搜索引擎的需要。至此，现如今的网页分类器模型需要不断的通过对特征的优化降低维度来减少机器学习模型的运行时间，或是对数据量进行大幅度裁剪使得机器学习模型所需要运行的数据规模变小以满足要求。

1.4 核心工作

本论文主要工作有，首先对本文所提出的网页分类模型构建方法的研究与确立进行详细描述，后分章节对传统的基于人工策略的网页分类器模型的开发设计和通过对不同机器学习算法的比较调研对基于机器学习算法的网页分类器模型的研究，本文还将提供一个集成多类策略模型的垃圾网页过滤系统。

本文提出的结合人工策略模型和机器学习模型的网页分类器模型正是出于对网页分类器开发中的数据规模大、准确率不高等主要问题的考量。通过第一阶段的基于人工策略的模型开发能够在相对于机器学习算法的极小的时间复杂度内先筛选出部分更为可能识别为指定网页类型的小部分网页集合，用于第二阶段的基于机器学习算法的网页分类器模型。换言之，人工策略模型主要用于裁减原始的巨大数据规模，机器学习模型在优化后的较小数据规模上加以识别分类，这样在时间效率及准确率上都得到保证。

1.5 结构安排

本文将在第二章首先介绍网页类型识别技术的研究现状及本文所提出的网页分类器模型的大致设计，对于本文所提出的网页分类器模型的设计主要包括的两类模型，其一是传统的人工策略模型，其二是基于机器学习分类算法的机器学习模型，将分别在第三章和第四章详细描述。本文第五章将会概述一个由多类策略模型所组成的网页过滤系统，其主要功能是利用其中各类型的策略模型对网页进行权重计算，将被识别为低质垃圾的网页过滤在网页库之外。

2 网页分类器模型构建方法的研究与确立

网页分类器模型分析的对象是单个网页或者多个网页[14]，是将传入分类器的 HTML 网页包分类成既定的网页类型，基本的方法是提取各种页面的基本特征（结构的或者语义的）使用机器学习算法或者人工规则进行组合，从而产生一系列的更高层的特征，如该页面是否属于索引页，该页面是否为低质垃圾页面类型等，供应用方使用。

2.1 问题定义

网页分类是对单个网页或者多个网页归类成之前已经预设定好的类别。分类是传统的有监督学习问题[6]，即一组已打上标记的数据用于训练一个未来可应用在标记实例的分类器。

网页分类的一般问题可以分为以下几部分更为具体的问题：主题分类，功能分类，情感分类，及其他类型分类[7]。主题分类是对一个网页的主要内容进行主题的归类，例如一个网页的主要内容是技术的讨论，则将其归为预定好的技术讨论类别；功能分类是对网页的用途进行归类，如若该网页的主要内容是博主写的博客则归为博客页，若该页面是论坛中的帖子则可归为论坛页等等。

基于问题预先分类的类别数量，可分为二类分类和多类分类，其中二类分类也就是将对象分成两种类别，而多类分类则是将对象分成多种类别，如图 2-1，本文中主要讨论的是多类分类。

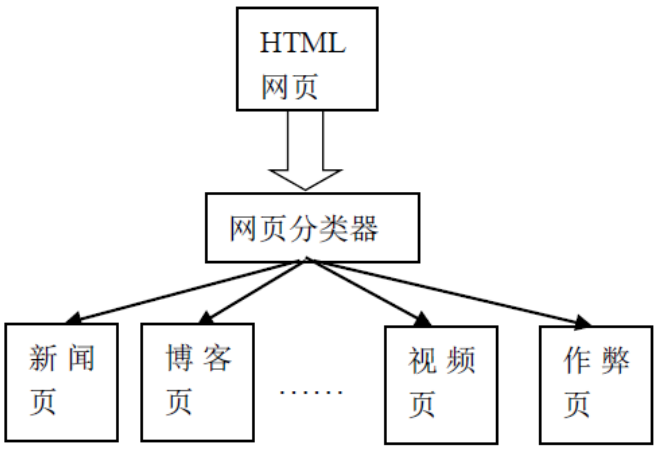

图 2-1 网页分类器工作原理图

由上图可看出网页分类是将一个 HTML 网页分成已经预先设定好的多种类型。目前的网页类型主要由人工根据需求来划分，一般的类型可以归到下面三个维度：结构维度、语义维度、结构语义双维度。

1.结构维度: 主要根据结构特征来划分网页，而不关心其页面内容。 典型的如索引页(列表页)与内容页。 例如，我们在识别列表页时，并不关心这个列表页是论坛列表页还是视频列表页，只要主体部分是列表，那么就是列表页了。

2.语义维度: 主要根据语义特征来划分网页，语义即网页中出现的词语。 典型的如色情页等等。例如，我们在识别色情页时，关键是看在这个页面中是否出现了色情相关词语，并不关心该页面的主题部分是大段文本还是列表，只要出现的色情相关词，并且够多够丰富，那么就是色情页面了。

3. 结构语义双维度：实际上，大多数的页面分类均需要根据结构和语义两方面的特征进行开发，典型的例如新闻详情页。从结构上讲，新闻详情页是一个内容页，需要与新闻列表页区分开；而从语义上讲，新闻详情页又需要出现新闻相关的词语。所以新闻详情页是一个结构语义双维度的页面类型。

下图是部分网页类型的举例。

1) 新闻详情页 NEWSCONTENT
2) 论坛帖子页 FORUM_POST
3) 视频播放页 VIDEOPLAY
4) 博客文章页 BLOGARTICLE
5) 商品详情页 PRODUCT_DETAIL
6) 视频列表页 VIDEOLIST
7) 色情页 SEX
8) 色情视频页 SEXVIDEOPLAY
9) 音频资源页 AUDIOPLAY
10) 下载资源页 DOWNLOAD
11) 图片页 PICTURE
12) 搜索结果页 SEARCH_RESULT
13) 索引页 INDEX
14) 黄页 YELLOW
15) 小说内容页 NOVEL_CONTENT
16) 小说章节列表页 NOVEL_CHAPTER
17) 小说首页 NOVEL_HOME
18)供求页 SUPPLY

图 2-2 网页类型实例图

在互联网中网页类型多种多样，无法枚举里面的全部类型。一些基础的网页类型，由我们提供的网页分类器模型开发的设计模式直接开发。而许多情况下，应用方还可以自己对页面类型进行定义，使用我们提供的网页分类框架进行网页分类模型开发，从而满足实际的页面数据挖掘需求。

2.2 传统的网页分类流程

对中文网页进行分类的一般流程主要是：首先对原始的 HTML 网页进行一些预处理操作，如 HTML 解析，词法分析，中文分词，停用词删除及词干提取等[8]，在对原始 HTML 网页预处理完成后，需要对已处理过的 HTML 网页进行特征抽取，或者叫做“上游计算”，也就是通过上游的计算模块对 HTML 网页进行必要的特征计算[15]，对已确定诸如 HTML 网页中含有的图片个数计算，HTML 网页中的有效文本长度，HTML 的网页结构等特征进行处理计算。对已抽取特征的网页传入网页分类器中进行分类。传统的网页分类的大致过程如图 2-3 所示.

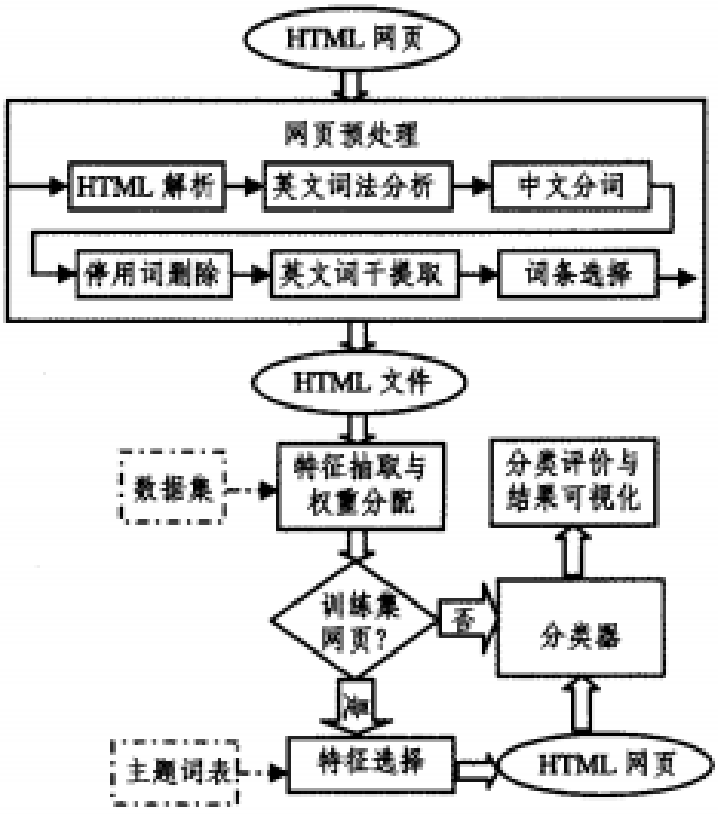

图 2-3 传统的网页分类流程

2.2.1 网页预处理

网页预处理就是将原始的 HTML 网页进行预先的处理生成一个易于进一步利用的浓缩的 HTML 文件。一般的网页预处理主要包括六个步骤：HTML 解析、英文词法分析、中文切词（也叫做中文分词）、删除停用词、提取英文词干以及选择词条。[3]

其中，HTML 解析主要目的是将 HTML 网页源代码中的有用信息（如各种标签的内容及属性值）提出，删掉无用信息（如 HTML 源代码中的注释部分）等；英文词法分析则主要是通过处理部分符号作为的分隔符，并将数字、标点、连字符等进行特殊处理，以及对各标签名及标签内容等进行小写化处理等；中文分词，又叫中文词法分析，传统的分词系统是将大量的词典加载后再对 HTML 网页中的正文内容进行分词分析。而我们的系统通过开发者自己创建词典进行分词，相比于加载巨大词典的传统的中文分词，开发者自己创建小部分的词典所需要的内存消耗可忽略不计，极大节省内存空间，该部分详细内容将在人工策略模型中加以说明；停用词删除主要作用在于删除那些出现频率过高不但对文本没有帮助，反而增加特征维度的词，这类词称为停用词。

2.2.2 特征抽取和选择

HTML 网页的特征主要体现在 HTML 源码中每个标签中的内容或是标签属性或是标签自身，以及 HTML 源码的结构[4]。特征的抽取，主要在网页分类器的前半部分，也就是我们所提及的基于人工策略的网页分类器模型的开发中，开发者人为地对可能有指定网页类型识别有关的特征进行抽取。如在 HTML 网页源码中，HTML 的结构包括：

主体（BODY），BODY 标签的内容。

元信息（META），META 标签下的元信息描述。

标题（TITLE）,网页的标题。

元信息标题(MT),元信息和标题组合的内容。[5]

在经过人工策略模型筛选后的网页进入机器学习模型之前,需要选择此前已抽取出的特征作为机器学习模型的特征空间构建机器学习模型。

2.2.3 分类器

这一小节的分类器不同于本文提出的网页分类器模型，其主要是基于机器学习算法的网页分类器模型（即此前提出的网页分类器模型的第二部分，机器学习模型）。

该分类器通过预先选定的机器学习模型，如最大熵模型，支持向量机模型等，作为我们网页分类器模型的第二部分，开发者抽取和选择的特征用于构建机器学习模型。在数据挖掘领域中，分类算法是解决对输入数据进行分类至既定的类型的分类方法。分类算法通过对已知类别的，即已经完成标注的训练语料集合分析，从中发现分类规则，以此预测新数据的类别。换言之，分类算法是一种有监督学习。在分类器模型开发过程中，机器学习模型的训练需要依靠大量已标注的数据进行训练。[9]经过多次多批量的数据

训练后，达到指标的分类器（诸如准确率高于 97%，特征维度不超过万级等指标）便可作为完整的分类器模型使用。

2.3 分类器模型的设计与构建

以往的分类器或是单纯用人工策略模型作为分类器模型，或是单纯使用机器学习模型，在开发周期和模型效果上难尽人意。单纯的人工策略模型开发周期长，由于因为人工策略所能考虑的特征种数有限，很难同时保证准确率和召回率符合搜索引擎的需要；机器学习虽然能同时考虑多维的特征，但是特征选择带有盲目性，且很难对已开发完成的网页分类器模型进行调试，由于机器学习模型是通过大量训练数据和多次的训练而构建的，如当出现将网页类型识别错误时，无法对已开发完成的网页分类器模型进行修改或难度成本极大。

为此，本文结合以往的人工策略模型和机器学习模型提出一种网页分类器模型的开发设计模式。该类网页分类器模型的大致设计结构如图 2-4。

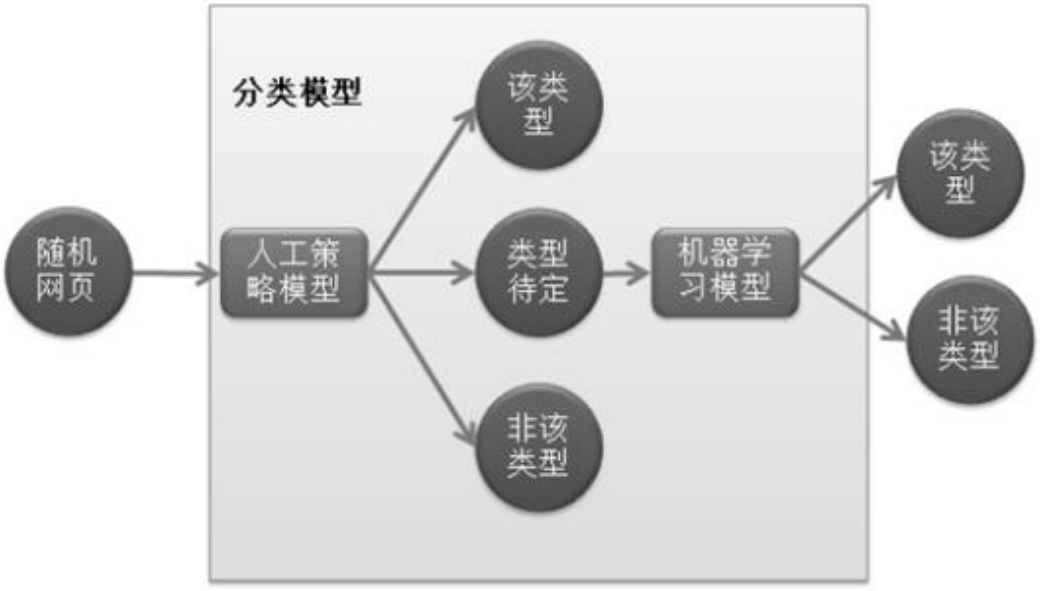

图 2-4 本文所提出的网页分类器设计结构

本文所提出的网页分类器模型的开发流程同其由两种分类器模型构成相同，需要分步开发，且如上一章节所描述，随机网页首先经过人工策略模型的筛选识别后，被确认为“类型待定”的网页将继续传入至机器学习模型进行过滤识别。在开发流程中，为了使各个阶段模型达到标准，符合搜索引擎需要，严格的评估贯穿着模型开发的始终，完

整的开发流程如图2-5。

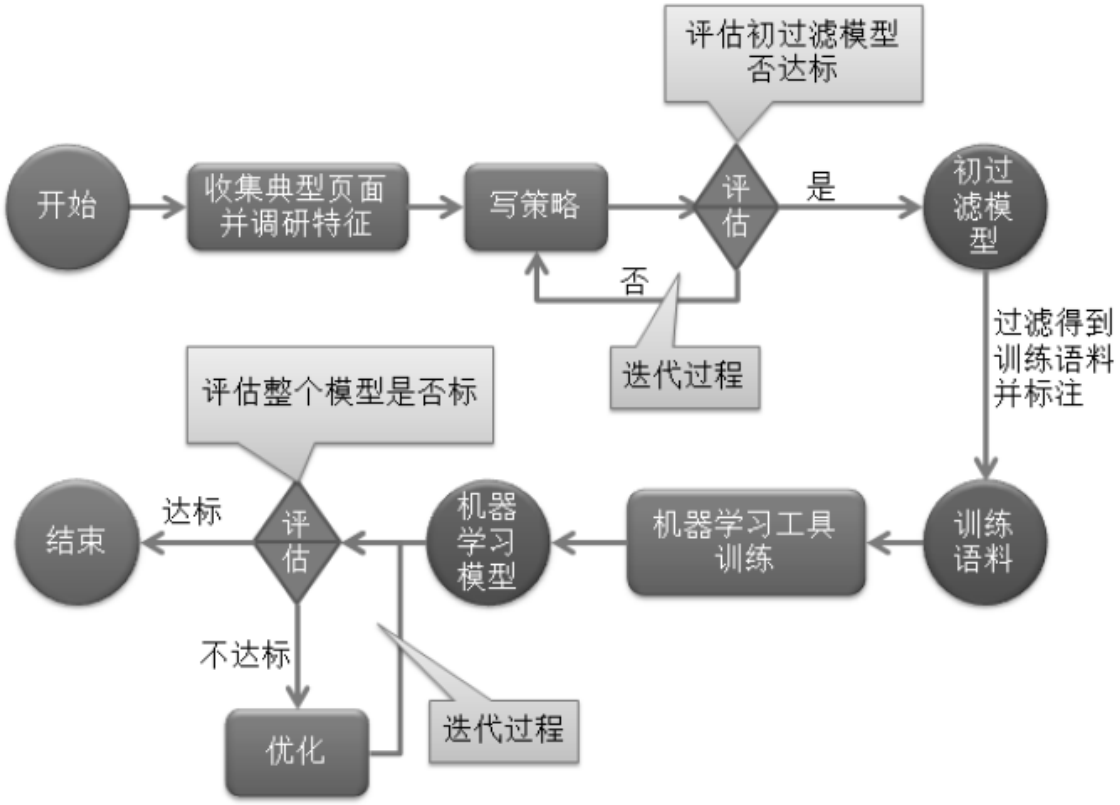

图 2-5 本文所提网页分类器开发流程

其中，“初过滤模型”即为开发者通过人为的选择特征并根据不同特征编写规则而成的人工策略模型；机器学习模型所需的“训练语料”的获得则是通过从离线网页库中随机抽取的网页进行人工策略模型过滤识别后“类型待定”的网页，再通过人工识别的方式对各网页进行标注指定类型。

基于人工策略模型（初过滤模型）的具体评估方法将在第三章基于人工策略的网页分类器模型中加以介绍。

分类器模型开发中训练语料的准备和标注的具体方法以及机器学习模型的构建将在第四章基于机器学习的网页分类器模型优选中详细描述。

2.4 与其他网页分类器的比较

2.4.1 传统人工策略的分类器模型

传统的人工策略模型是由模型开发者人为的制定多种策略，后对各种策略进行组合或综合打分，最后指定规则决策网页的分类。如图 2-6 所示。

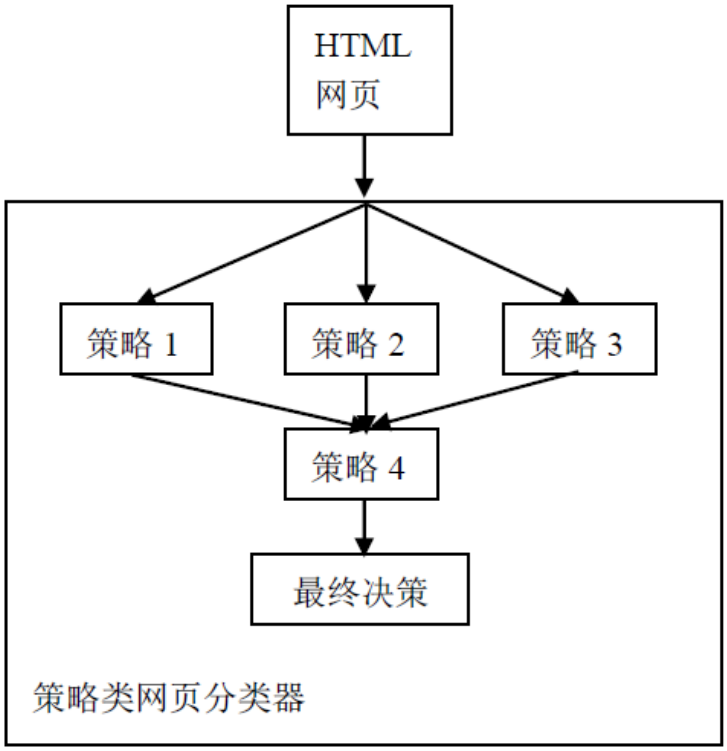

图 2-6 人工策略模型简化图

优点：时间复杂度极小，远低于基于机器学习模型的网页分类器。

缺点：准确率较机器学习模型低，足够优化的人工策略模型的网页分类器准确率在80%左右。在当对网页分类的准确率需求极高的时候（99%以上），该类模型无法满足要求。如当搜索引擎需要识别并过滤垃圾网页的时候，仅用准确率 80%的人工策略模型，将会导致近五分之一的非垃圾网页无法被搜索引擎收录，极大影响搜索引擎的覆盖率。

2.4.2 基于机器学习的分类器模型

基于机器学习的分类器模型，即是通过应用成熟的机器学习模型，对网页进行分类。该类分类器模型如本章第二章第二节所描述的传统的网页分类类似。其主要工作在于数据的准备，预处理和数据标注上的准确率在很大程度上决定了最终模型的准确率和召回率。

下图 2-7 为机器学习模型中用于分类的模型构建步骤，图中可看出，对于基于机器学习模型的网页分类器模型的构建主要在于训练数据的准备（预处理、标注等）。

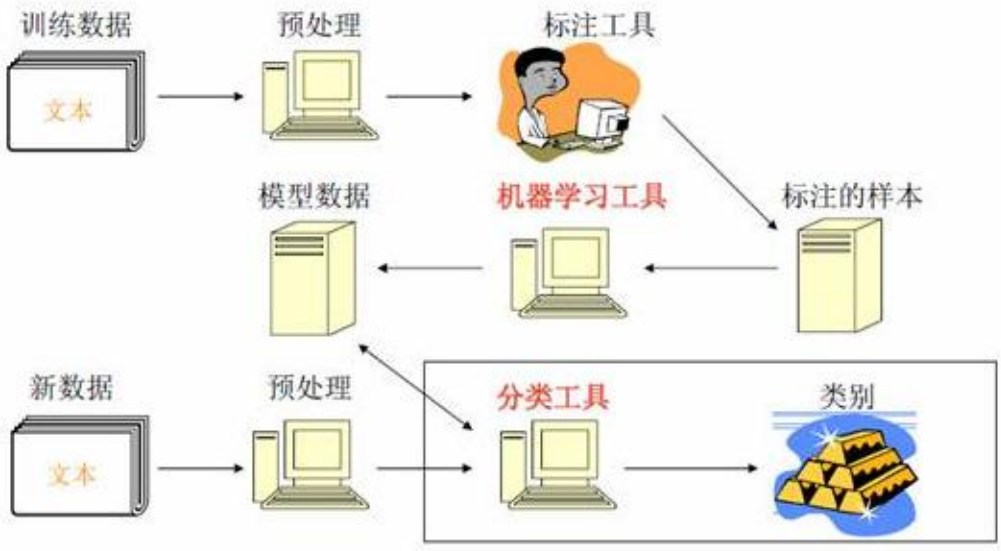

图 2-7 用于分类的机器学习模型的构建流程图

机器学习模型分类器的简化图如下：

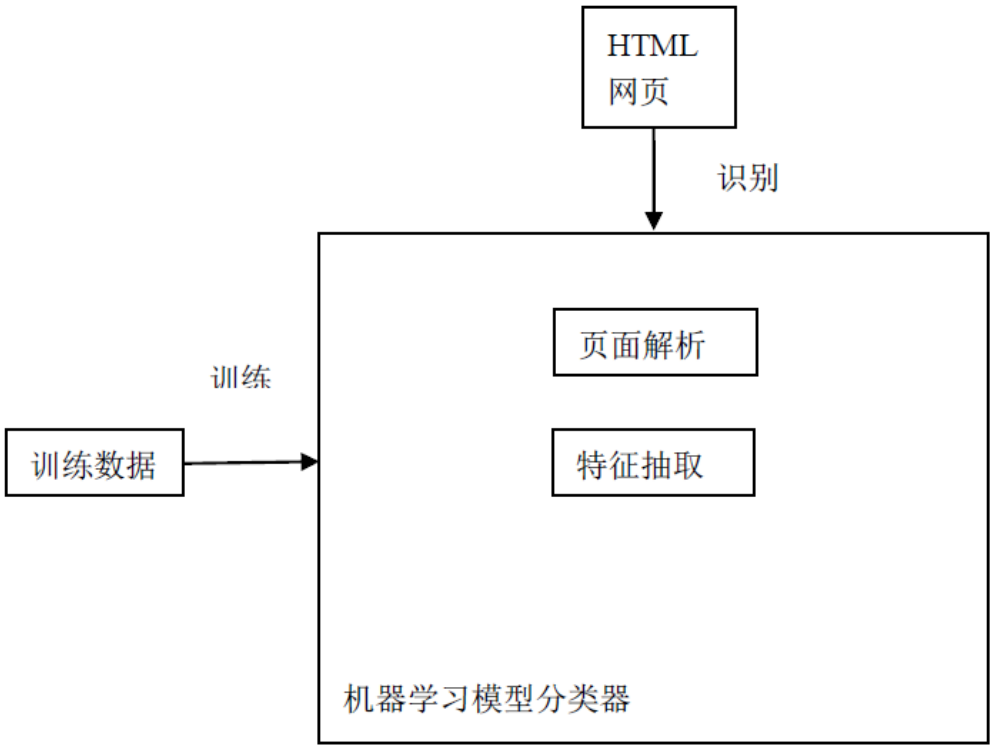

图 2-8 机器学习模型简化图

优点：充分训练和优化后的机器学习模型具有极高的准确率，基本满足搜索引擎对分类的准确率需求

缺点：训练数据的准备需要大量人力及时间；特征维度的增长导致机器学习模型分类器的运行效率极低。

2.4.3 结合人工策略和机器学习的分类器模型

本文所提出的网页分类器模型中将人工策略模型作为第一部分子模型，机器学习作为第二部分子模型。我们将人工策略模型和机器学习模型当作我们网页分类器模型连续不可分的两部分，其设计的主要目的是通过人工策略模型过滤大量非指定类型的HTML网页包或已确定属于指定类型的HTML网页包。

如图2-9为该类分类器模型的构建模式

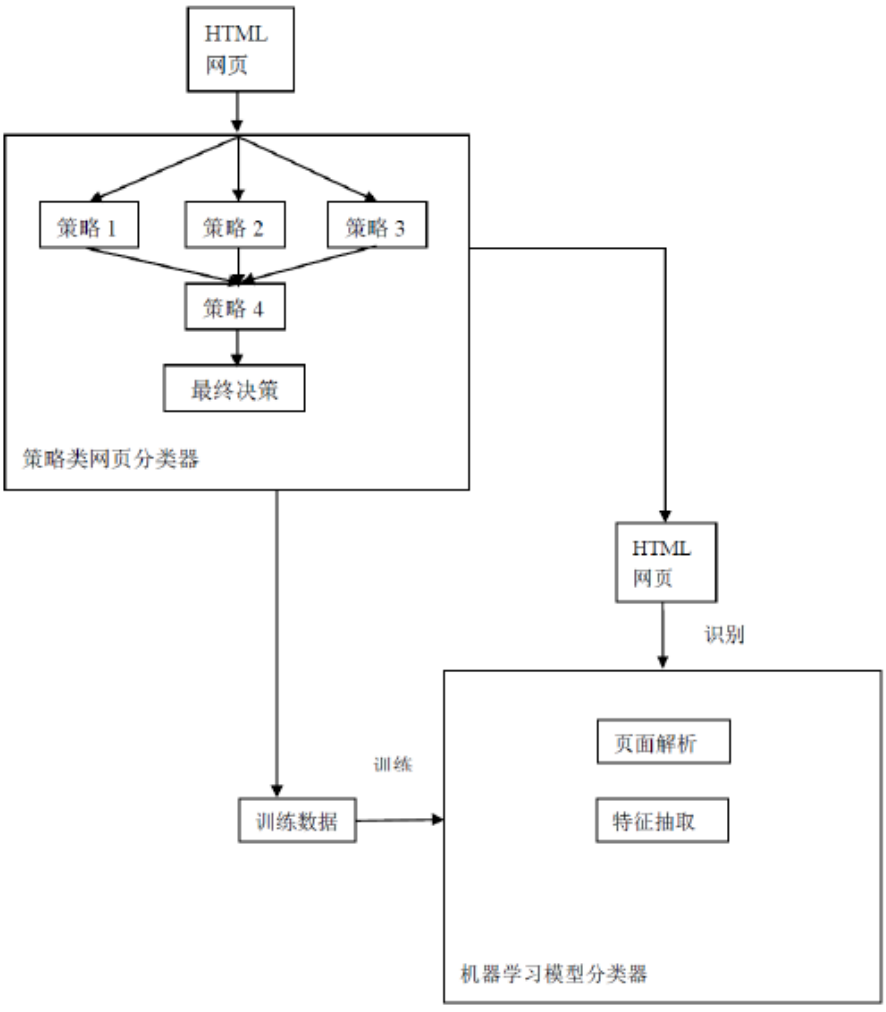

图2-9结合人工策略和机器学习的分类器模型简化图

这样的结合人工策略模型和机器学习模型的网页分类器并不是简单的将两类模型合并。实际上，两类模型甚至不能独立使用和独立构建。比如第二部分的机器学习模型的训练数据需要通过第一部分的人工策略模型，其目的是为了让训练出的模型更具针对性。

以下为该中网页分类器的主要设计思路及原因:

(1) 机器学习模型无法解决全部问题。

如果想让一个单纯的机器学习模型达到满意的分类效果，则势必要求机器学习的各个环节都调试到最好：算法的选取，特征的选择、语料的分布、参数的选择等，这极大的增加了开发模型的难度，如训练语料的标注准确度以及需要巨大的训练语料数量。而人工策略带有很强的目的性，并且大部分典型的页面特征明显，人工策略很容易区分。因此人工策略模型解决一部分最典型的HTML网页，机器学习模型再解决已通过人工策略模型的部分HTML网页，从而极大提高准确率，且可以与人工策略模型互相弥补不足。

(2) 人工策略可以调节训练语料的类型比例

理论上，我们希望训练语料越多越好，但实际语料标注常常是机器学习模型开发的时间瓶颈，它虽然简单，但是十分耗时，需要人工一个个看网页。许多类型在随机网页中的分布很稀疏（比如视频页通常在随机网页中占比不到1%），如果我们想得到含有100个正样本的语料集合，需要标注10000个样本，代价太大。

引入人工策略模型可以较好的解决这个问题。我们开发一个高召回低准确的人工策略模型，比如召回率100%，准确率40%，然后把通过人工策略模型的样本作为训练语料，这样标注1000个语料中，期中正样本就有400个。并且实践证明，以现有的特征，开发一个高召回低准确的人工策略模型并不困难。

(3) 人工策略模型可以补充机器学习策略的不足。

机器学习产出的模型很难以调试，对于单纯的机器学习模型，如果我们发现某些特别典型的页面识别错误，除了增加学习语料和调试参数，没有更好的方法。而如果加了人工策略，可以很容易地将它们召回。

(4) 人工策略模型算法的时间复杂度极低，可以提高模型的整体性能。

使用人工策略，我们很可能利用一两句if/else语句就过滤掉大批网页。我们尽量将较为简单的策略放在整体策略的最前方，较为复杂的策略(如机器学习模型)放在整体策略的后面，这样可以使得性能高的语句优先计算，提高了模型整体的计算性能。

在极大规模大数据量（由第一章所提及的库中现存1400亿的网页数量）首选通过人工策略模型（初过滤模型）的初级筛选后，将筛选后的相比于此前大数据量而言极小的数据集合再通过传统的基于机器学习模型的网页分类器筛选识别，以提高准确率至搜索

引擎的需要。

3 基于人工策略的网页分类器模型构建

人工策略就是开发者从原始的HTML网页中抽取开发者想要的认为可能对筛选识别指定网页类型有帮助的网页特征，并通过已抽取的特征人为的制定规则对传入进来的HTML网页包进行筛选。在我们的开发系统中，开发者使用C/C++或是lua等编程语言进行人工策略模型的开发。在本章中将会涉及到人工策略模型的开发目的和意义，模型的开发流程，模型的开发技术，如特征的抽取与选择及模型开发完成后的评估方式。

3.1 模型设计流程

在确定所开发的网页分类器模型需要筛选识别的指定网页类型之后，开发者首先需要收集典型页面，并调研相关特征。典型页面也就是指定网页类型的典型的页面，例如，当前开发的网页分类器模型需要能够识别出新闻页，则开发者可以通过收集新浪、网易、凤凰网等新闻门户网页。之后通过开发者对典型页面的调研后，在人工策略模型之中进行策略开发者认为的可能对模型识别指定网页类型有帮助的特征，或者说是相关性强的特征进行特征的抽取和选择，并最终制订合适规则用于筛选网页。

图 3-1 是人工策略模型的开发流程。

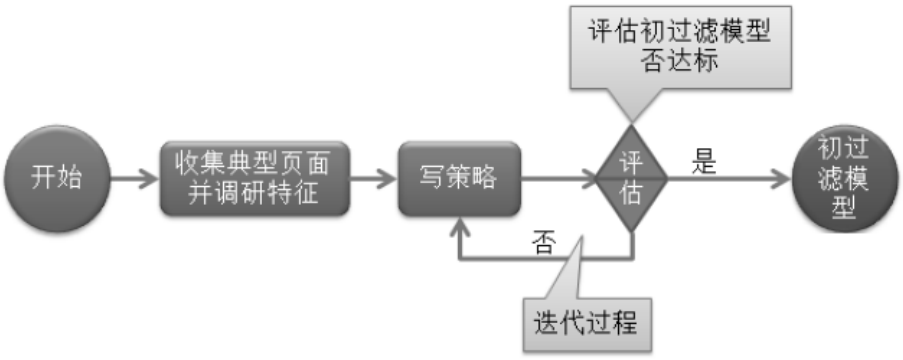

图 3-1 基于人工策略模型的网页分类器开发流程

人工策略模型开发初期的典型页面调研需要开发者收集并调研少量的（通常约为500 个）指定类型的网页。人工策略模型开发的关键步骤在于开发者不断的修改人工策略并通过抽样数据进行评估，直到初过滤模型（人工策略模型）达到评估标注后，即产出初过滤模型。

3.2 典型页面收集

典型页面的收集方式主要包括以下两种:

1)通过query在搜索引擎中收集：

使用目标类型query(比如视频页就是视频query)，在google.com,360 搜索,baidu 等搜索引擎中进行搜索。例如，取query数量100,三个搜索引擎的前10的结果，共得到*3000url.list*，需要将其去重，预计得到的HTML网页包的数量约在1000至3000范围内。将*3000url.list*送至PM标注，最后将标注为指定类型的网页作为典型页面集。

2)PM 直接在互联网中搜索，或通过类似 hao123 入口页进行人工的类似广度优先搜索的方式进行指定网页类型的搜索和收集。

典型页面收集过程产出：

得到 100 个站点，每个站点 5 个对应页面，共 500 个典型页面。

对于典型页面，随机分为 3 份。

1)第一份:200个，包括40个站点，设为*X_recall.pack*。

2)第二份:100个，包括20个站点，设为*Y_recall.pack*。

3)第三份:200个，包括40个站点，设为*Z_recall.pack*。

其中*X_recall.pack*、*Y_recall.pack*、*Z_recall.pack* 供下节评估方法中使用。

3.3 特征选择

人工策略模型效果的好坏关键在于开发者所抽取的特征是否准确、选择的特征与识别指定页面类型网页是否具有相关性，以及人为制定的规则是否准确且能够很好的平衡准确率与召回率。

在我们的模型开发平台中已提供较全面的页面级特征的抽取接口，下图为开发平台所提供的部分页面级网页特征，页面级特征是对指定的特征进行整体页面上的计算，计算出的特征值供开发者设计策略。

```
enum PageFeaType
{
    //url_fea_func
    UrlPathLength,                  //URL path的长度
    UrlWord,                        //URL 切词
    UrlPathWord,                    //URL path 切词
    UrlSiteWord,                    //URL site 切词
    UrlDateEnd,                     //URL 是否满足日期结尾
                                    //http://neaapostole.wordpress.com/2009/10/08/
                                    //http://neaapostole.wordpress.com/2009/10/
    UrlStandardDateContentEnd,      //URL 是否满足日期+内容结尾
                                    //http://blog.educastur.es/pompas/2013/05/23/doki-descubre-el-agua/
    UrlStandardEnd,                 //URL结尾是否是个内容
                                    //estiman-que-la-construccion-crecio-18-en-junio
                                    //mario-poli-sera-el-arzobispo-de-buenos.html
    //htmltag_base_fea_func
    HtmlTag,                        //HTML tag名组成的wordlist,例如div:5,a:6
    HtmlAttr,                       //HTML attr名组成的wordlist,例如id:10,name:1
    TagCombineAttr,                 //tag与attr组合的wordlist,例如div_class:9,meta_name:10
    TagCombineAttrValueLength,      //tag与attr组合,attr中的属性的属性值长度的wordlist,例如div标签的class属性的值全部长度为100
    TagTextLength,                  //tag下某标签文本的长度例如<p>aaa</p> p下文本就是3
    TagCombineAttrNumber,           //tag与attr组合的个数
    //htmltag_wordlist_fea_func
    TagInfoValue,                   //tag中的属性的属性值切词
    TagInfoText,                    //tag下的文本的切词
    AllText,                        //全部文本节点切词 例如包含<script>**</script>之间的文本
    PureText,                       //全部纯文本(text node)切词 ，不包含例如 <script>,注释文本等
    StylePureText,                  //全部纯文本加上样式的切词
    NotAnchorPureText,              //纯文本中非锚文本的切词
    XPathInfo,                      //Xpath组合例如<div><h1><b>aaa的Xpath为div.h1.b.#text
```

图 3-2 网页分类页面级特征

上图是部分的页面级特征及相关说明举例，此处举几例加以说明：

1) UrlWord

如果开发者想知道，网页 url 中是否含有有“/video/”字符串，可以请求计算 UrlWord_/video/特征，计算结果会返回该页面 url 中含有/video/字符串的个数，如 UrlWord_/video/:1

2) TagInfoValue

表示指定 vnode 结点指定标签中的关键词个数，由于是页面级特征，会将所有的个数合并。举例来说，在做视频播放页识别时，发现 div 标签下的 id 属性中如果含有 video 词，表明该 div 块常常是加载播放器。

3) TagInfoText

与 TagInfoValue 类似，但是匹配的是指定标签的文本匹配情况，而并非属性文本匹配情况。

3.4 规则制定

人工策略筛选规则的制定，就是开发者通过已抽取的特征，对量化后的特征单独地设置阈值或是先对已抽取的特征进行组合。人工策略模型主要是由两部分构成，其一是特征的抽取，也就是对 HTML 源码进行相应的计算，或是通过平台给出的页面级特征和接口通过平台系统实现底层计算后直接提供给开发者；其二则是通过已抽取的特征组合而成的一条条策略。通常的做法可以是每条策略对传入的 HTML 网页进行筛选，得出结果是否为指定类型网页，是否为待确定的指定类型网页；或是每条策略对传入的 HTML 网页进行打分，最终通过得分设定阈值加以分类。具体如图 3-3。

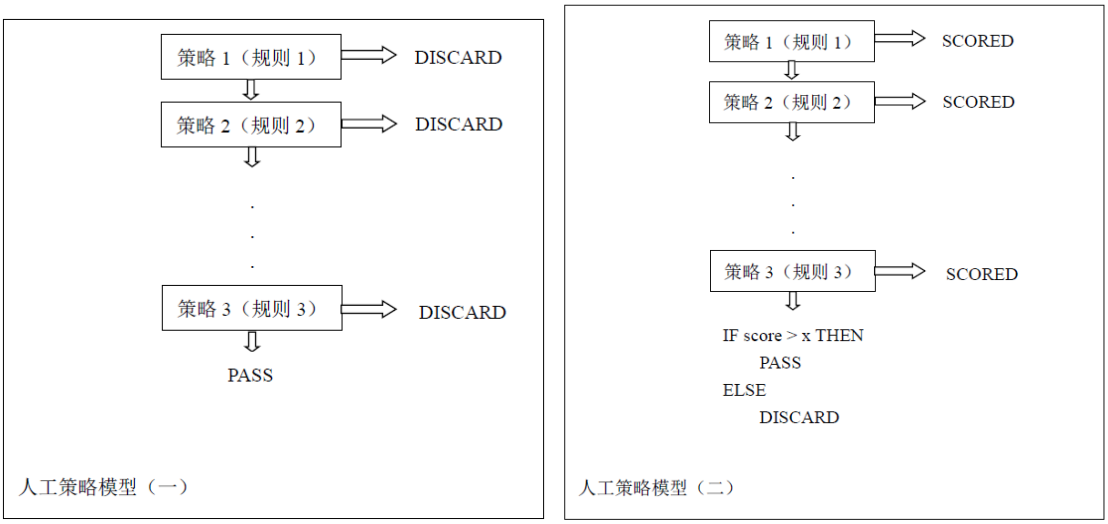

图 3-3 人工策略模型的基本形式

3.5 评估方法

该阶段的评估方法分成“自评”与“送评”两种，分别由开发人员自己评估与送至 PM（产品工程师）处进行评估。

首先在离线网页库中随机抽取网页包，在不同的小库中抽取 3 个每份数量为 10w 的随机网页包，分别设为 *A_rand.pack*、*B_rand.pack*、*C_rand.pack* 作为人工策略模型迭代开发的抽样数据。

自评：

使用 *A_rand.pack* 与 *X_recall.pack* 进行初过滤初始开发。

准确率:随机抽取 *A_rand.pack* 中的 HTML 网页的识别准确率达到 60%以上。

召回率:*X_recall.pack* 典型网页包中通过人工策略模型所召回的网页达到 95%以上。

送评：

使用 *B_rand.pack* 与 *Y_recall.pack* 进行初过滤模型(M0 模型)的迭代开发。

将使用模型 *M0* 过滤 *B_rand.pack*，将通过模型的网页，随机抽取 1000 个，送至 PM 处标注。

获得标注样本 *P0*。

准确率迭代:使用 *P0* 迭代 *M0*，尽量剔除 *P0* 中被标注为 0 的页面（即识别错误的 HTML 网页），且尽可能减少损失标注为 1 的页面，确保召回率不减。

召回率迭代:使用 *Y_recall.pack* 测试召回率，达到 95%+。

3.6 人工策略模型的实际构建——论坛帖子页

本节将详细介绍实际的网页分类器模型的构建，将以开发论坛帖子页为例从模型构建最初的明确定义至编写策略再到抽取选择特征及最后的策略模型评估等实际构建流程描述。

3.6.1 明确定义

在确定开发指定某种类型的网页分类器之后，首先，应该清楚论坛帖子页是怎么定义的，什么样子的网页才是论坛帖子页。只有通过人工能判断出来某随机网页属于指定类型网页之后，才有可能将其编写为实际的网页分类器模型让程序去识别。所以第一步自然是开发者首先熟悉了解自己所要开发的网页分类器期望识别的网页类型的常见特征。比如，对于论坛页，论坛帖子页指的主要是 BBS 的具体帖子页面，通常为不同的网友编辑而成。

3.6.2 编写策略

在本章之前已经阐述了基于人工策略的网页分类器模型实际上就是由非常多条策略（或者说是规则）进行组合后构建而成的策略模型。所以，开发基于人工策略的网页分类器模型在实际开发中最重要的就是设计编写每条子策略，并且要求开发者能够自己评

估每条策略的优劣。通过将具有不同准确率和召回率的多条策略组合构建成一个策略模型就是我们所开发的人工策略模型。下面主要描述论坛页的人工策略模型开发。

假设开发者已通过挖掘 50 个站点近 200 个论坛帖子页的典型页面后，已经发现了如下一些常见特征（这类常见特征都由开发者自身从现有数据中挖掘出）：

1）页面为一楼一楼的回帖(Post)组成，通常情况下回帖由不同网友编辑，在每层楼的侧面位置(有时也在顶部)一般都会有回帖者的用户简介。如图 3-4。

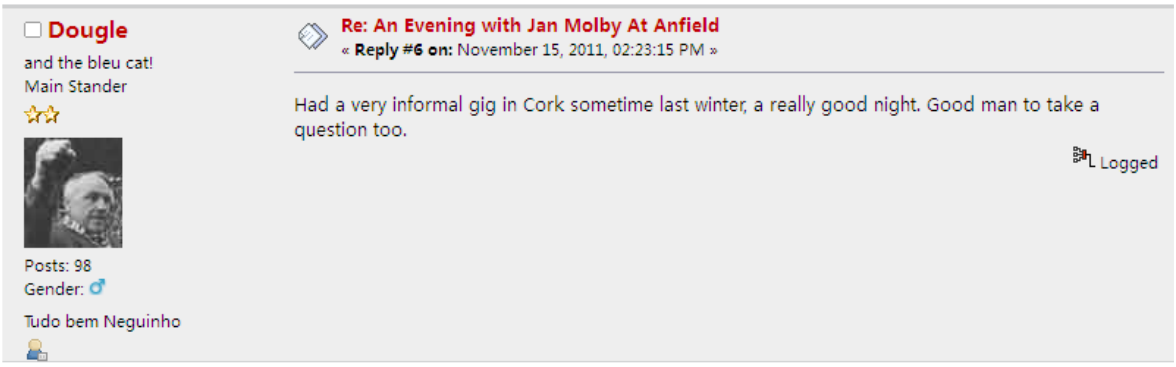

图 3-4 论坛帖子页常见特征（二）

2）通常页面的每一个 Post 的下面都有引用(Quote)和回复(Reply)的按钮，便于用户参与论坛讨论，如图 3-5.

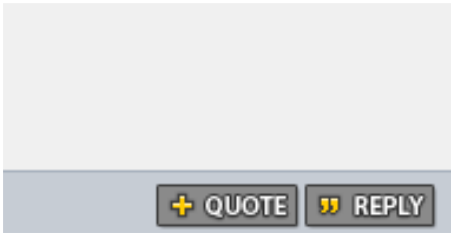

图 3-5 论坛帖子页常见特征（三）

3）页面主体的上部或者下部还有发帖(New Topic)的按钮，点击后发表新的主题帖，如图 3-6

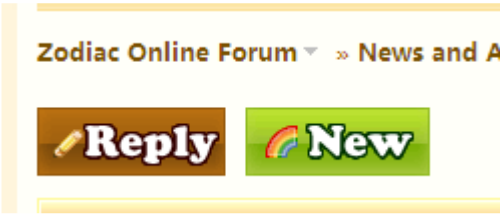

图 3-6 论坛帖子页常见特征（一）

4）url 所带特征，如论坛帖子页的 url 中带有/forum/,/foro/,/forum.php,/topic/等。

5）标题中所带有特征，title 标签中的文本带有“论坛”，“帖子”（根据各国语言）等相关词。

以上仅列出 5 条，实际开发中的人工策略正是通过以上这样的一条条特征转换成策略规则而构建而成的人工策略模型。

3.6.3 抽取特征

在开发者挖掘发现出论坛页的诸多特征之后，需要将这些特征从 HTML 网页的源码中将重要信息抽取出来，这时便要求开发者通过现有开发平台所提供的接口或是自己对 HTML 网页源码的解析后进行抽取，最后再将抽取出的信息（大多数情况下会是量化后的特征值，HTML 网页所含的图片个数，url 路径中是否含有 forum 切词等）通过特定语言（本项目中使用 C/C++）将其编写成程序后便是该人工策略模型。

3.6.4 策略评估

由于并不是每条策略都是有实际收益的策略，或者是达到指标要求的策略，如有的策略准确率不够，或是召回率不够，都有可能严重影响整个人工策略模型准确和召回率。所以在开发者每挖掘并设计编写完一条策略后，便需要对该策略所召回的 HTML 进行抽样调研该条策略的准确率与召回率，已确保模型最终效果。当整合多条子策略的时候，开发者不仅要平衡准确率与召回率，还要保证整体模型的运行效率在可运行的条件下。

4 基于机器学习的网页分类器模型优选

本文第二章中提及的，我们所提出的网页分类器模型设计模式的第二部分是一个完全基于机器学习算法的网页分类器模型。本章主要讨论如何应用有指导的机器学习方法实现网页分类器模型的设计。本章将从机器学习模型的设计开发目的及意义说起，并介绍常用于网页分类的成熟的机器学习算法，后选择其中的最大熵模型(Maximum Entropy Models)和支持向量机模型(Support Vector Machine)等机器学习算法及不同的优化方法进行实验比较，最终确定我们所选择的用于我们机器学习模型的机器学习算法，本章的最后将会介绍我们机器学习模型的开发流程包括特征的抽取选择和机器学习模型训练语料的准备过程。

网页分类是搜索引擎中的一个古老的技术话题，由于搜索引擎技术的不断快速发展，搜索引擎对信息检索技术的要求也不断提高[17]。对网页分类所要达到的目标也不断提高。如今真正能够上线应用的网页分类器模型所需要的准确率标准也不断提高要求至97%甚至更高。之前所介绍的基于人工策略的网页分类器模型的主要目的在于降低数据规模提高网页分类器模型整体的运行效率，但同时牺牲了准确率，经实践发现，单纯的人工策略模型的准确率最高至 80%左右，远不能满足搜索引擎对模型准确率的需求。所以本文提出的网页分类器模型的设计中需要通过如今发展较为成熟的机器学习模型来构建网页分类器使最终的网页分类器模型的整体准确率能够达到标准。总而言之，网页分类器中的机器学习模型的目的在于在已降低数据规模的情况下极大提升网页分类器的整体准确率，弥补对人工策略模型设计开发中开发人员对网页特征选择的盲目性和局限性。

4.1 基于最大熵模型的网页分类器

最大熵定理，也称作最复杂定理。在物理学的领域中，物理学家们用一概念来描述客观事物的复杂程度，这个概念即是熵。所有的客观事物自动地使自己内部状态的复杂程度在所有已有的约束条件下使其能够达到最大值。要在所有现存约束中选择能够使得该分布通过现有的所有约束条件且不能确定唯一的一种分布时，已经经过证明的便是“最好的分布就是最复杂的分布”，也就是熵值最大的分布。[10]

4.1.1 最大熵模型原理综述

在我们的基于机器学习模型的网页分类器中，最大熵模型是我们采用的其中一种机器学习模型，本节讨论最大熵模型在网页分类中应用的理论基础。

在对 HTML 网页进行指定类型识别的问题中，一个事件可以看成是将传入进模型的HTML 网页分到某个指定网页类型，在之前已抽取和选择的特征中，如 HTML 页面中出现的某些特定的词或 HTML 网页的结构等信息可以看成是将该网页分成某指定网页类型这一事件发生的环境，我们想得到包含这些特征的 HTML 页面属于该指定的网页类型的概率，可以通过如同在朴素的贝叶斯算法中那样统计训练语料来确定。

定义 C = {c1, c2, … , cn}是对网页分类之前所预先设定的网页类型(Category)的集合，F = {f1, f2, … , fm}是进行网页分类时所选取的特征(Feature)，训练语料中，即已标记的数据中，二元组（ci, fj）表示人为判定为ci类型的HTML网页中包含fj特征的出现次数。则有如公式4-1：

$$P(ci,fj) = \frac{(ci,fj)}{\sum_{i=1}^{n}\sum_{j=1}^{m}(ci,fj)} \qquad (4\text{-}1)$$

然而，存在多数 (ci,fj)在训练数据中没有出现的情况时，将会极大影响P(ci,fj)的结果，在朴素贝叶斯算法中可以通过拉普拉斯平滑方法解决以上概率为零的情况。然而在最大熵模型中，它使未知事件的概率分布总是尽可能的平均，即尽可能得到最大熵[18]。

4.1.2 网页分类中的最大熵模型

假设，现在对一个随机网页进行类型识别，且仅通过 HTML 网页源码中抽取的文本信息并进行切词后匹配到词典中已有的词语“基金”，然而通过训练语料得知含有这个词的文本信息的网页只可能是新闻页、博客页、论坛页三种网页类型，那么则有公式 4-2:

$$P(c1 \mid b) + P(c2 \mid b) + P(c3 \mid b) = 1 \qquad (4\text{-}2)$$

其中 P(c1|b)，P(c2|b)，P(c3|b)分别表示出现“基金”一次的文本信息的网页分类至新闻页、博客页、论坛页的概率。在未知其他约束的情况下，最大熵模型则尽量使这三种类别的概率相等，则有公式 4-3：

$$P(c1 \mid b) = P(c2 \mid b) = P(c3 \mid b) = \frac{1}{3} \qquad (4\text{-}3)$$

但是通过对模型的训练，及训练语料的分析与统计，得出该类网页分为新闻页的概率达到 60%，那么将网页分类至其余两种类型的概率便会被认为 20%。

最大熵原理可以表达为：从一系列允许的概率分布p(a，b)中选择具有最大熵H(p)的模型 p’，该模型具有最均匀的分布[4]，即:

$$p' = \arg\max_{p \notin C} H(P) \qquad (4\text{-}4)$$

4.1.3 最大熵模型优选的实践验证

该调研实验目的在于利用现有的网页训练语料和特征抽取工具，评估最大熵模型在网页分类上的效果，并进行相关模型的参数调试实验。本次调研同时也对不同的机器学习优化方法进行对比，如随即梯度下降(sgd算法),改进的拟牛顿法(L-BFGS算法)等.

本次调研实验的原始数据为空短报错页面类型训练语料，训练集 4588，测试集 919，正样本比例 80.84%，特征抽取完毕且配置好切词词典后，提取到的特征维度为 21488.

训练输入数据格式如下：

label f1 : w1 f2 : w2 f3 : w3 ... fn : wn

- label 为类别标记，在二分类中为 1 或-1
- f 为特征的 id，可以为正整数(最小特征 id 为 1，偏移值默认 id 为 0)，也可以为字符串类型
- w 为特征的权值，为整型或浮点型

字符串类型特征如下图：

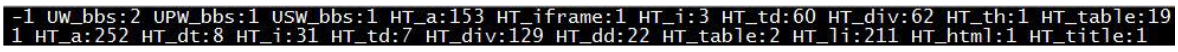

图 4-1 字符串类型特征示意图

整数类型特征如下图：

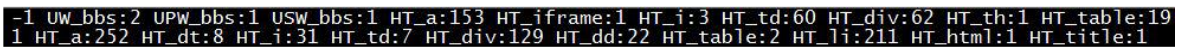

图 4-2 整数类型特征示意图

在网页分类中，我们主要关注 label 为 1(正样本的) 准确率，因此，通过比较不同参数训练下模型的测试结果中的 label 为 1 的 precision(并结合 recall)，来进行参数调优。

（以下为特征未进行归一化的结果）

迭代次数为[30, 200]，仅给出正样本召回率 80% 以上的最优结果如表 4-1：

表 4-1 最大熵实验数据表（未归一化）

编号	优化算法	lambda	gamma	precision(%)	recall(%)	迭代次数
1	lbfgs	0.01	0.005	90.3520	92.7711	59
2	lbfgs	0.005	0.0025	90.5138	91.9679	61
3	lbfgs	0.001	0.0005	90.6824	92.5033	62
4	lbfgs	0.0005	0.0001	90.4575	92.6372	57
5	lbfgs	0.0001	0.00005	90.5836	91.4324	113
6	sgd	0.01	0.005	93.0448	80.5890	157
7	sgd	0.005	0.0025	92.0777	82.4632	141
8	sgd	0.001	0.0005	91.8836	80.3213	199
9	sgd	0.0005	0.0001	92.2961	81.7938	188
10	sgd	0.0001	0.00005	91.0581	81.7938	197

对特征经过归一化以下公式 4-5 处理：

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4-5}$$

迭代次数为[30, 200]，仅给出正样本召回率 80% 以上的最优结果如表 4-2：

表 4-2 最大熵实验数据表（已归一化）

编号	优化算法	lambda	gamma	precision(%)	recall(%)	迭代次数
1	lbfgs	0.01	0.005	89.3140	90.6292	64
2	lbfgs	0.005	0.0025	89.9204	90.7631	197
3	lbfgs	0.001	0.0005	89.6053	91.1647	103
4	lbfgs	0.0005	0.0001	89.9729	88.8889	199
5	lbfgs	0.0001	0.00005	89.8817	91.5663	44
6	sgd	0.01	0.005	81.2840	100	-
7	sgd	0.005	0.0025	81.3725	100	33
8	sgd	0.001	0.0005	85.6476	98.2597	70
9	sgd	0.0005	0.0001	90.0253	95.4485	52
10	sgd	0.0001	0.00005	90.6372	93.3066	51

4.2 基于支持向量机模型的网页分类器

本节将介绍我们用于开发基于机器学习算法的网页分类器模型的机器学习算法，支持向量机（Support Vector Machine，简称SVM），是一种机器学习的方法，在统计学的不断发展过程中，是通过统计学的理论基础上发展而来的一种机器学习方法,其主要基于结构风险最小化原理，将原始数据集合压缩到支持向量集合,学习得到分类决策函数[5]. 支持向量机模型的主要思想是构造出一个决策平面，常为一个超平面，使正负模式之间的空白最大。经实践发现支持向量机在文本分类等方面表现良好，主要体现在分类的召回和准确都好于现有的大部分方法。

4.2.1 支持向量机模型原理综述

在分类问题中，为了易于理解，常可以将数据及分类方式形象地映射至一个二维或三维的平面，这样有利于对分类算法的理解。然而在实际上，数据绝大部分都含有多维的特征，所以实际上的分类若要映射至平面上也应该是个“超平面”，超平面常指三维平面以上的平面，由于三维以上的平面无法用现实世界中的实物来模拟描述，所以用超平面来描述时也较为抽象。这里同样的，我们在描述支持向量机模型原理时也将其简化至用二维平面加以描述。

首先如下图，可以很明显的看出，图中通过一条直线，将两类不同的（圆形和方形）形状划分成了两部分（直线的上下两部分）。在分类算法中，图中的那条直线实为一个线性判别函数，又叫线性分类器[19]。

线性判别函数，从定义上说，这是一个由 x 的各个分量的线性组合而成的函数：

$$g(x) = w^T x + w_0 \qquad (4\text{-}1)$$

当 g(x) ＞ 0 时，则该判定函数将 x 分至于类 C1

当 g(x) ＜ 0 时，则该判定函数将 x 分至于类 C2

当 g(x) = 0 时，则可将 x 任意分至于其中一类或是不对 x 进行判定。如图 4-3

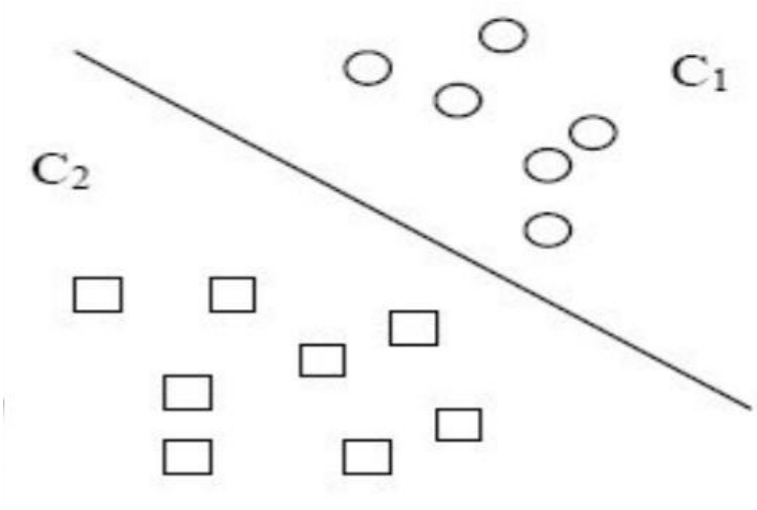

图 4-3 二维线性判别函数图

以上是线性分类，然而 SVM 是从线性可分的情况下的最优分类面所发展而来的，支持向量机的主要思想可用图 4-4 的二维平面加以说明

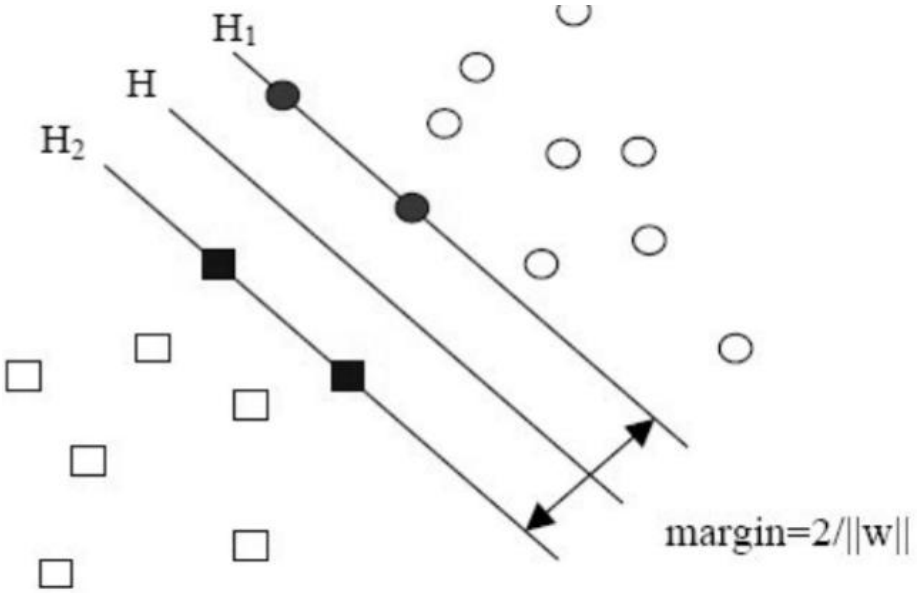

图 4-4 二维支持向量机原理图

图中，正方形点与圆形点分别表示不同的两个种类样本，分别设其为类别 C1 和 C2，线 H 为分类线，H1、H2 分别为过类别 C1 和类别 C2 中距离分类线最近的且平行于函数 H 的直线，他们之间的距离又叫做分类间隔(margin)。在支持向量机模型中，最优化分类线也就是要求分类线 H 不仅能将两类正确的分隔开，而且同时还能使得其分类间隔最大[20]。

支持向量机模型所要做的就是，对训练数据不断的训练之后，找到这样一个分类间隔并使其最大化。支持向量机模型在确立最大分类间隔面的时候其最终的决策函数一般只通过所有数据中较少数的支撑向量所确定也就是说其计算的时间复杂度主要取决于训练数据中的支撑向量的个数，而非样本空间的特征维度[21]，这在我们之前提及的 HTML 网页类型不同于传统文本分类时的特征多样化中就避免了特征维度的扩大导致的模型效率低下。

4.2.2 网页分类中的支持向量机模型

在基于机器学习算法的网页分类器模型中，支持向量机模型需要对传入的训练数据以及配置好的特征进行训练，模型主要完成的是二元分类，即可简单的由 4.3.1 节中的图表示，可将圆形点看成是被支持向量机模型划分为指定类型的 HTML 网页，正方形点则为非该指定类型的 HTML 网页。

具体的训练过程主要包括：准备数据、选择特征、提取特征、训练模型、模型调优等步骤。

1） 准备数据：

由于所训练的支持向量机模型只需要对传入的 HTML 网页包进行二元分类，即是否为指定类型网页，所以在准备数据阶段中，只需要对数据进行 0 和 1（或-1 和 1）标注分别指代网页是否为指定类型网页。标注实例如图 4-5 所示：

```
 1 1   http://www.diypda.com/thread-51827356-1-1.html
 2 1   http://www.irzbmj.dplffb.cn/bbs/thread-225938-1-1.html
 3 1   http://www.v84.dectop.cn/
 4 1   http://share.renren.com/share/601044867/14214598048?fin=9&expose_time=1403953766
 5 -1  http://auto.cnhubei.com/data/news/20120709/story_139426.html
 6 1   http://www.ma7i0.cn/news/2015102299987.html
 7 1   http://www.jx.cn/jixie/1304/1jcpgyview.asp?id=58911797&l3=隆子&l2=南&l=香港
 8 1   http://ilkpfa.cn/2015/10-24/515nr/2015/0807/296098.html
 9 1   http://ndxzwb.com/tags/fulnr.html
10 -1  http://bbs.081868.com/thread-107833-1-1.html
11 1   http://www.cheyuan.com/vip/88113399/index.php?mod=cars&id=283187
12 1   http://www.ebay.de/cln/chcn591/2015-02-24/160015617012
13 1   http://m.123du.cc/dudu-34/53386/130718.html
14 1   http://www.293.net/www.fortney.com
15 1   http://bbs.cfan.com.cn/bcw0818/35080530.html
16 -1  http://cz.caihao.com/rsmd-15041802896797.html
17 1   http://zx.zynews.com/qwjd/keyword/xnkbhdxyyy.html
18 1   http://bbs.dm123.cn/read-htm-tid-57542179.html
19 1   http://www.exam8.com/gwyzwxxk/4_岑溪市做假职称资格证办假职称资格证%2B343630627-QQ号65x/
20 1   http://www.diypda.com/thread-19054766-1-1.html
21 1   http://www.wdzfw.com/elenalwawiw/
22 1   http://c90i.diypda.com/thread-55755563-1-1.html
23 1   http://tj.esf.sina.com.cn/house/f3-a9-t1-c187,229/
24 1   http://wap.020mk.cn/chapter/ctx_1494184.html
25 1   http://www.jinnong.cn/tradesdhtmyjt/2014/5/2/20145222261057368.shtml
26 1   http://www.cstfbo.com/ebbaeefc/19.html
27 1   http://www.echinagov.com/news2015092260861/
"train_data.label" 4831L, 267358C
```

图 4-5 支持向量机模型训练数据示意图

2） 选择特征：

在对数据标准准备完成之后，需要选择相应特征对支持向量机模型的训练，下图为实际开发中对支持向量机模型训练之前的特征选择和特征配置。如图 4-6 所示。

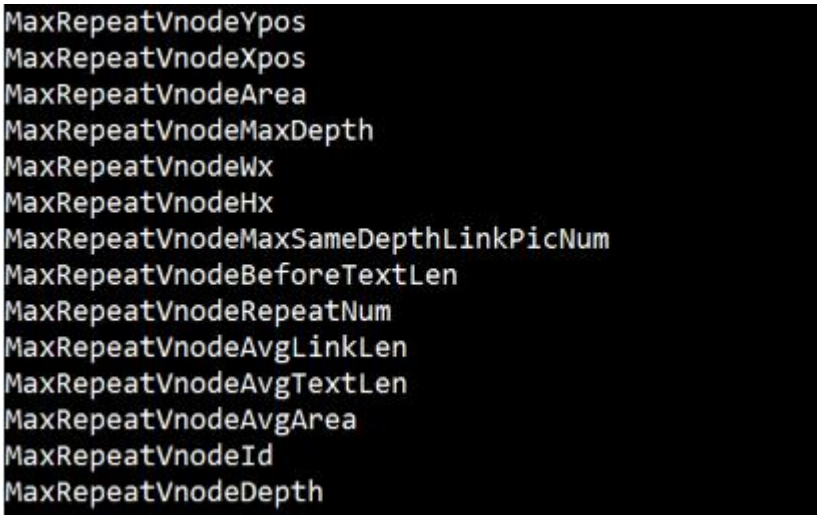

图 4-6 支持向量机模型特征配置示意图

3） 特征提取

在选择完相应特征之后，需要使用相关工具对 HTML 网页包进行特征提取，图 4-7 为特征提取完成后的文本状态

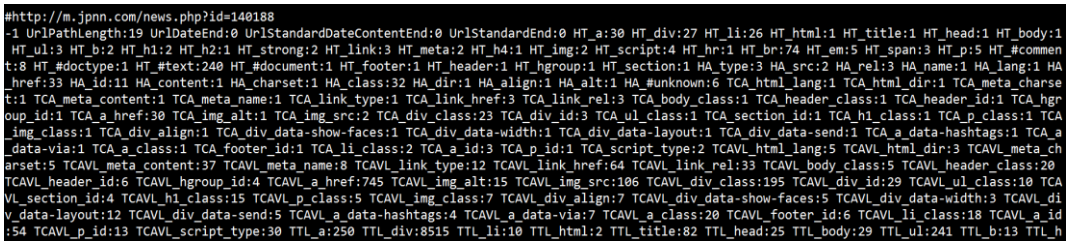

图 4-7 特征提取原始数据示意图

4） 训练模型及模型调优

在训练支持向量机模型的时候，不同的训练数据，就可能需要不同的优化方法和不同的模型迭代次数，通过枚举迭代次数，并在其中较好的几组模型中衡量模型准确率和召回率，最后选取一组作为最后支持向量机模型的最终模型。在下一节的调研实验中，将会具体描述这一过程，并给出实验结果。

4.2.3 支持向量机模型优选的实践验证

本次调研实验的原始数据为空短报错页面类型训练语料，训练 4588，测试集 919，正样本比例 80.84%，特征抽取完毕且配置好切词词典后，提取到的特征维度为 21488.

训练输入数据格式如下：

label f1 : w1 f2 : w2 f3 : w3 ... fn : wn

- label 为类别标记，在二分类中为 1 或-1
- f 为特征的 id，可以为正整数(最小特征 id 为 1，偏移值默认 id 为 0)，也可以为字符串类型
- w 为特征的权值，为整型或浮点型

训练输出结果如图 4-8 所示：

```
instance: 4588
feature-value pairs: 1358237
all features are digits!
uniq feature number:20678 (include b)
Total y class number: 2 : 1 -1
using sgd for training....
objective: min avg max{0,2-(<w_y,x>-max<w_k,x>)} + 1/2*0.0005*||w||^2
sampling without repalcement
iter_num  accuracy    objective  = Hinge_Loss + L2_reg
================================================================================
1         76.3296% 6.352497   = 5.509414   + 0.843082
2         84.8954% 1.821875   = 1.318333   + 0.503543
3         89.0802% 1.060631   = 0.717241   + 0.343390
4         90.5623% 0.800777   = 0.528729   + 0.272048
5         92.2842% 0.660668   = 0.428560   + 0.232108
6         93.1125% 0.574758   = 0.370354   + 0.204404
7         93.1997% 0.530308   = 0.345492   + 0.184816
8         93.9407% 0.480145   = 0.309446   + 0.170699
9         94.1369% 0.459973   = 0.299019   + 0.160955
10        94.9869% 0.425985   = 0.272581   + 0.153405
11        95.4228% 0.412362   = 0.263648   + 0.148714
12        95.1177% 0.394488   = 0.250951   + 0.143537
13        95.4882% 0.385061   = 0.244773   + 0.140288
14        95.6408% 0.365749   = 0.229364   + 0.136385
15        95.7934% 0.361296   = 0.227888   + 0.133408
16        95.9024% 0.356720   = 0.225379   + 0.131340
17        95.9459% 0.343173   = 0.214788   + 0.128386
18        96.0549% 0.341579   = 0.215512   + 0.126067
19        96.0549% 0.332915   = 0.209219   + 0.123696
20        96.1203% 0.326937   = 0.205575   + 0.121362
finished training.....
Time to train: 0.4 seconds.
```

图 4-8 支持向量机模型调研实验数据示意图

- 关键输出信息说明
  - *instance* ： 训练集的规模
  - *uniq feature number* ： 包含偏移特征的特征维度，实际特征维度为该值减1
- 输出模型文件，以*model_file*为*model*为例
  - model.bin ： 二进制模型文件

- model.txt ： 文本模型文件（最终作为SVM 模型文件集成到网页分类模型中）
- 若采用了特征映射(-m 参数)，则还有对应的odict 文件: model.ind1, model.ind2, model.n

特征为原始数据抽取的特征，未进行归一化时，多次调整参数训练均不收敛。

对特征进行线性归一化处理后

迭代次数为[30, 200]，仅给出正样本召回率80% 以上的最优结果，如表4-3：

表 4-3 支持向量机实验数据表（已归一化）

编号	l(L2 正则化系数)	precision(%)	recall(%)	迭代次数
1	0.01	86.5636	97.4565	97
2	0.005	89.0954	96.2517	33
3	0.001	90.9922	93.3066	33
4	0.0005	91.3793	92.2356	36
5	0.0001	92.6031	87.1486	38
6	0.00005	92.4286	86.6131	52

表中，precision 代表模型的准确率，recall 代表模型的召回率。通过实际的调查与研究，并权衡准确率和召回率的需求，最终在已有的模型中确定最优的模型。

5 过滤系统的设计与实现

本章将介绍我们项目中通过开发网页分类器模型进行筛选识别垃圾网页所在的过滤系统，该过滤系统属于搜索引擎 spider 下计算中心的子系统。本章将分别从该过滤系统的设计背景、系统功能、设计需求及目的、整体框架、详细设计与实现等方面对该模块进行描述。

5.1 设计背景

搜索引擎通过爬取系统（crawler）从互联网中爬取最新的网页后，需要对网页进行各种方面（维度）的计算，以便之后对刚爬取的网页进行各种操作，如建库、索引、排名等都需要根据上游模块所计算出的属性或是特征，使得最终用户在使用搜索引擎搜索感兴趣或是想要的信息的时候，搜索引擎能够更准确更快捷更全面的为用户呈现用户所希望得到的信息。

本章所介绍的过滤系统即可看成是对网页进行入库，索引，排名操作之前的上游计算模块。该过滤系统亦可在原始的 HTML 网页刚被搜索引擎的爬取系统抓取后，通过该过滤系统可确定是否将该 HTML 网页存入网页库等功能。

5.1.1 系统功能

过滤系统（Filter system,FS），是一个通过多种策略模型对 HTML 网页包进行页面分维度打分并最终将各个维度整合的系统，维度可以是不同的页面类型，也可以是页面的不同的考量方向，比如从网页是否属于稀缺网页的稀缺程度，或者网页的总体页面质量等，还可以是 HTML 网页的 URL 链接的属性，该网页站点来源的权威性等，每个维度分别计算得分(score)和置信度(confidence)，最终将各维度的得分和置信度做类似加权平均的整合，得到页面最终的得分。

5.1.2 系统背景

该过滤系统是资源价值实时判断的策略框架和实现模块，负责质量维度和部分受众维度的价值判断，并拟合出最终的检索价值打分，用于指导垃圾过滤和索引分层。

下图 5-1 是该过滤系统所在的整体系统背景。

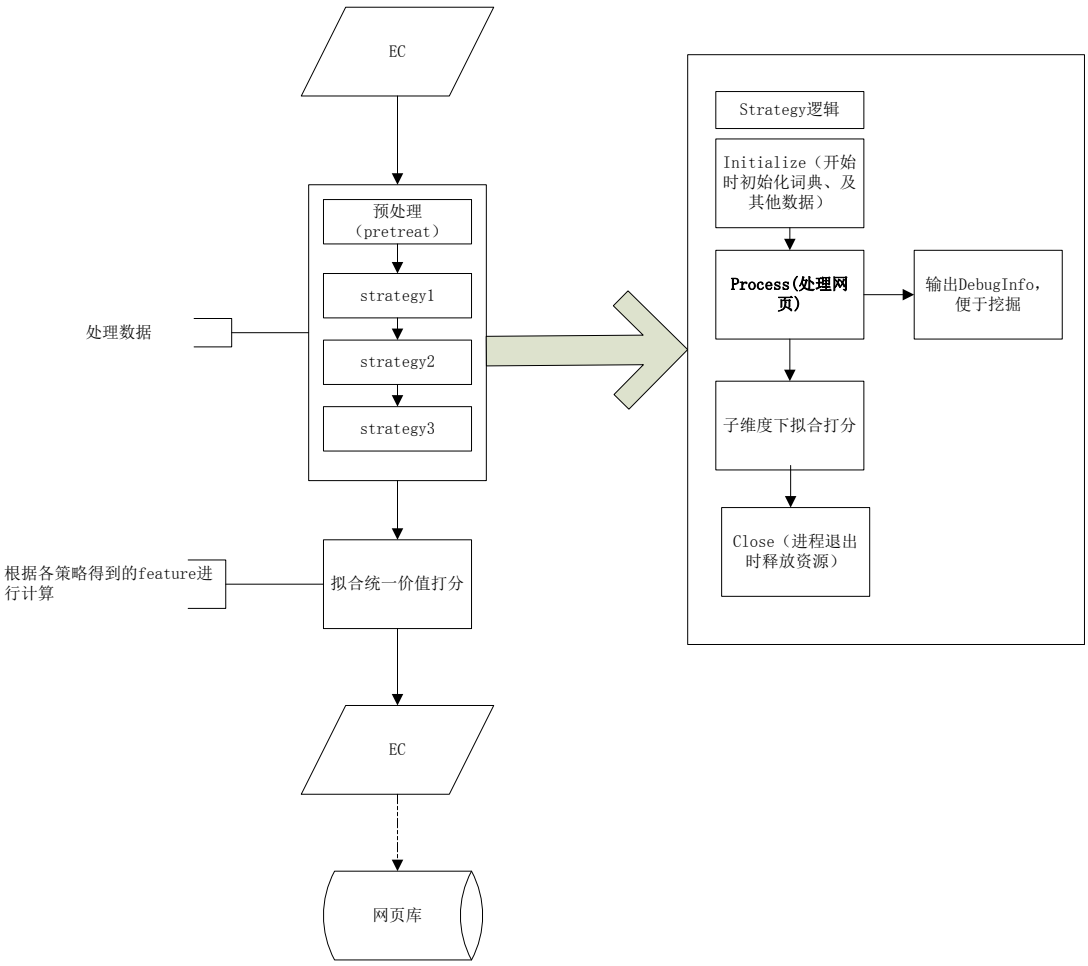

图 5-1 过滤系统的系统背景

图中 EC 全称为 Extraction Center 模块，是网页采集系统 Spider 的一个模块，其功能主要可分为三大类分别为提取、识别分类以及页面变化判断。其中提取功能包括提取网页建库需要的标题、超链、正文等信息，提取 EC 计算页面类型需要的中间结构信息；识别分类功能包括特定文件类型识别、索引页类型识别、页面有效性识别、页面语义分类等；页面变化判断为页面调度周期提供基础数据。在 HTML 网页通过 CS（crawler system）系统从互联网中爬取下来后，将传输至 EC 系统进行入库前的网页信息挖掘。

5.1.3 名词解释

以下是部分与该过滤系统相关的名词解释：

1)页面价值/质量，页面价值和页面质量实际上是相近的概念。页面的价值主要由三个因素构成：页面内部价值、站点价值和超链价值。对于低端，侧重于前者，更接近质量这个概念。对于高端，侧重于后两者，更接近于价值这个概念。但也不绝对，比如一

个普通站点上没有什么反链的网页，是个博客页，有很多的评论，那么它也具有很高的价值。本文后面将混用两者。

2)分数(score)，即对传入进系统的网页进行各维度的价值打分（特征向量计算后的整合打分）。FS 系统可以对传入的 HTML 网页进行打分，并输出[0,3]的分数，其中分数为 1 的网页为低质网页，分数为 0 的网页为垃圾网页，因此可以通过开发模型对网页打分，将网页 score 标记为 1 或 0 的网页从网页库中删除，达到过滤垃圾网页的作用。

4)置信度(confidence)，即对于对每个维度打分的可信程度。可视为准确度。

5)子模型(aspect)，也称为子策略，比如图片方面价值、视频方面价值、博客方面价值、文字方面价值、商品方面价值。即，页面内部价值是由多种方面的价值组成的。

5.2 设计需求与架构设计

该节主要从系统的功能性需求和非功能需求对该过滤系统进行描述。

5.2.1 功能性需求

以下是该系统的功能性需求：

1） 该过滤系统能够对传入进该系统的 HTML 网页包进行特征向量计算并产出在不同种类的页面间可比的且与询问无关的页面价值分数(score)，其分值符合与 PM 制定的 0—3 分标准，如下图表。

	一般页面	blog	色情页
3分	高检索价值、无作弊 (wiki, youtube, 重要主页,丰富资源页索引页等)	名人blog、且质量较高	资源丰富、无作弊
2分	大多数普通页面	质量一般的名人blog, 或资源较多/有检索价值的普通blog	有作弊但资源丰富, 或有一定资源且无作弊
1分	低质量、检索价值不大（bbs水贴等）	质量一般、检索意义不大的普通blog	资源不多、有作弊现象
0分	无检索价值（包括卖完页、死链、纯作弊等）		

图 5-2 网页类型分类原则

2） 该过滤系统还可将对 HTML 网页已计算出的价值分数中分数为 0 的垃圾网页或分数为 1 的低质网页进行过滤，即防止该类网页进入公司网页库。

3） 该过滤系统还应能够输出更丰富的数据，包括每个子维度（质量维度、受众维度等多维度）的价值分数、置信度、及其特征向量的计算。下图是页面价值的泛资源价值体系说明，也是对各个子维度的需求涵盖。

泛资源价值是在前面提到的资源检索价值三维打分中扩展出来的，从更多角度衡量资源的检索价值。下图 5-3 详细的表明了泛资源价值的各方面。

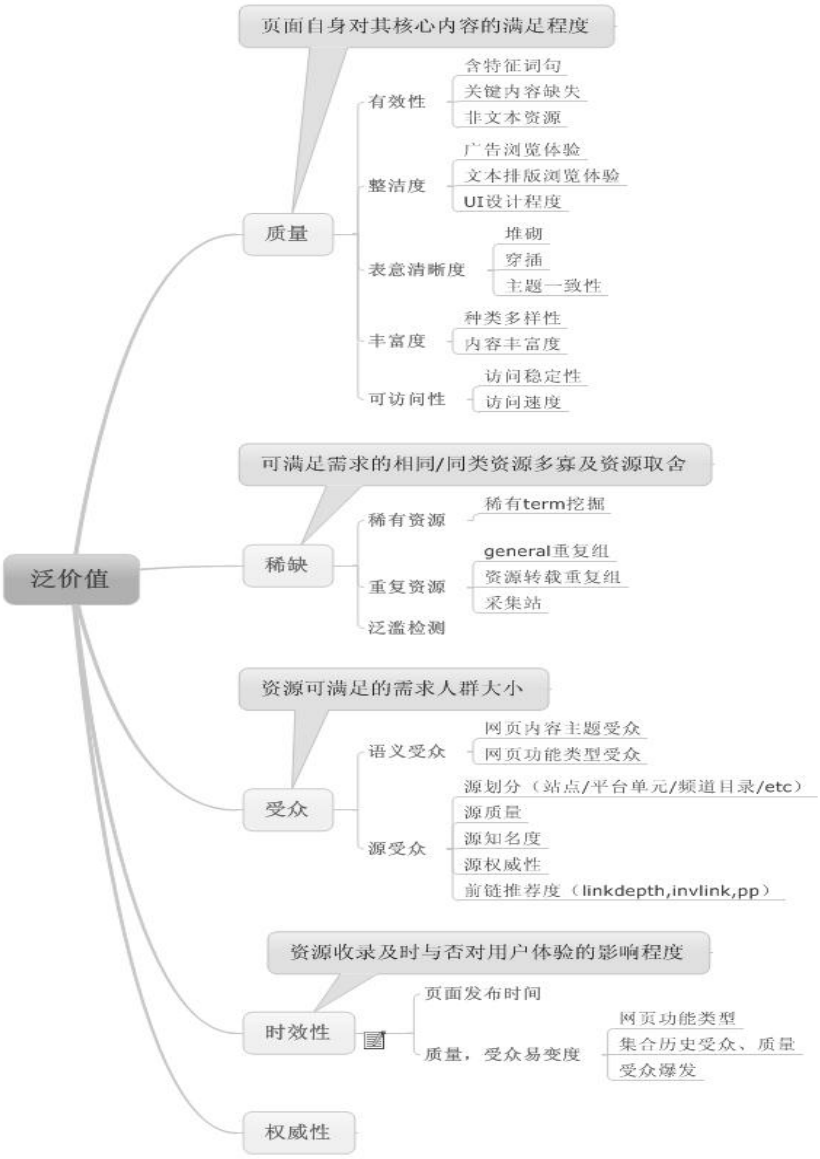

图 5-3 泛资源价值维度分布图

该过滤系统主要从泛资源价值的多个维度对传入的 HTML 网页进行特征计算和打分。对多个维度（如质量维度、稀缺维度、受众维度等）的划分，将过滤系统中的各类策略进行了划分，以及对每个子维度下该过滤系统所应能够根据其特点计算出的特征值等。以下是对于各个子维度的功能性需求：

1）质量维度：质量维度衡量页面自身对其核心内容的满足程度。从网页自身内容出发，判断对其所阐述的主题或对象是否清晰、明白，简单的说，就是如果用户需要知道该主题的详细内容，那该网页内容是否能满足需要。

(1)有效性维度：

该维度主要从网页信息的各个能够满足用户（网页浏览者）需求的子元素信息是否有效存在，根据表现形式不同，具体的在过滤系统中的模型划分中可以划分为过期信息页模型，空短报错页模型（空页面、主题短小页面、报错页、权限页等）等，以及非文本资源主体有效性四个方面，如图 5-4。

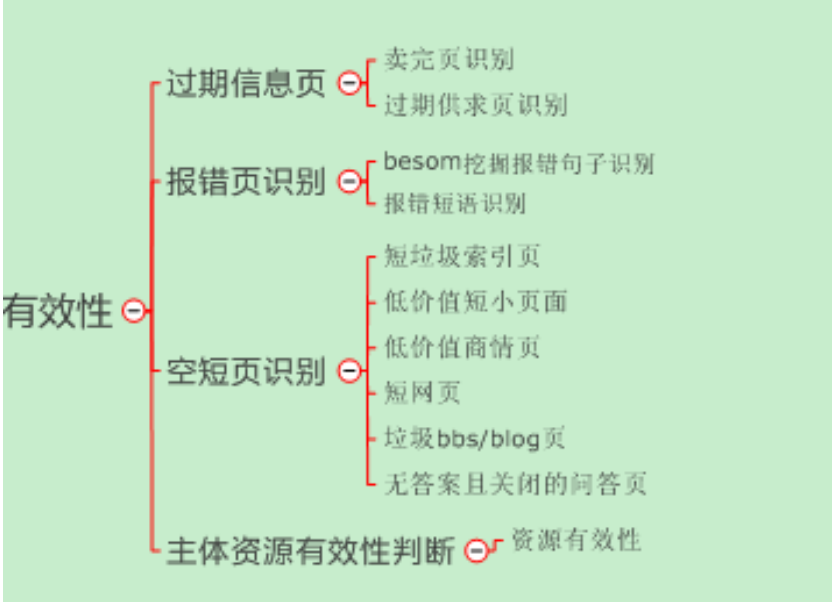

图 5-4 有效性子维度

(2)整洁度：

整洁度主要关注用户（网页浏览者）在 HTML 页网页的结构布局方面的满意度，如广告体验，结构排版，UI 设计等特征进行综合考虑并计算特征值。如图 5-5.

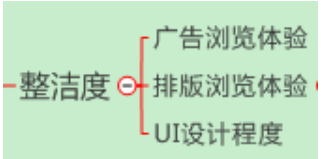

图 5-5 整洁度子维度

(3)表意清晰度

表意清晰度主要关注页面内容在语义上的表达，是否表意清晰，即能否易于用户（网页浏览者）理解等，如主题是否明确，垃圾信息或者广告信息等与主题无关的信息的堆积穿插是否明显等特征进行综合考虑并计算特征值。如图 5-6

图 5-6 表意清晰度子维度

(4)丰富度

丰富度主要关注页面所涵盖的信息量是否丰富。可以将其分为种类多样性，内容丰富度两类。实际计算中，丰富度判断主要依赖页面级特征的提取，比如，图片的识别和信息抽取是否准确等，同时需要结合页面分类才能拟合出有区分度的打分。如图 5-7.

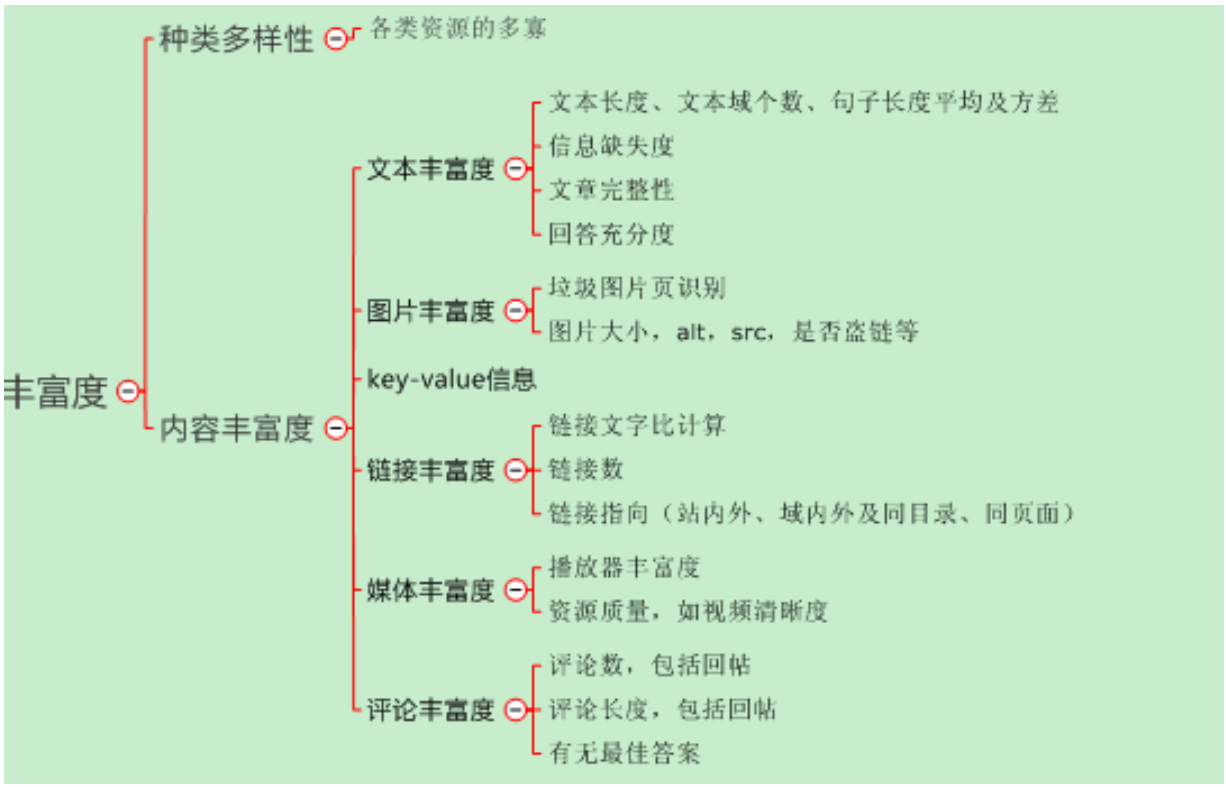

图 5-7 丰富度子维度

(5)可访问性

可访问性主要关注网页是否属于权限页，即是否需要登录、注册等权限才能访问的网页。对不可访问的页面，即权限页应能够识别并标记为低质网页。

2）受众维度：受众衡量资源可满足的需求人群大小，可满足的需求人群越大，受众越高，页面就越重要。目前受众维度的特征较少，思路还不清晰，该过滤系统中只纳入一些能够反映受众大小的简单特征来拟合打分，待受众维度的特征完善后再细化和扩展。

受众维度又分以下两种子维度，分别为语义受众和源受众：

(1)语义受众

受众衡量资源可满足的需求人群大小，可满足的需求人群越大，受众越高，页面就越重要。目前受众维度的特征较少，思路还不清晰，pv 中只纳入一些能够反映受众大小的简单特征来拟合打分，待受众维度的特征完善后再细化和扩展的受众大小，不同类型的页面本身就有受众差异（根据互联网用户行为和习惯确定），可以根据这点来打压垃圾。如图 5-8.

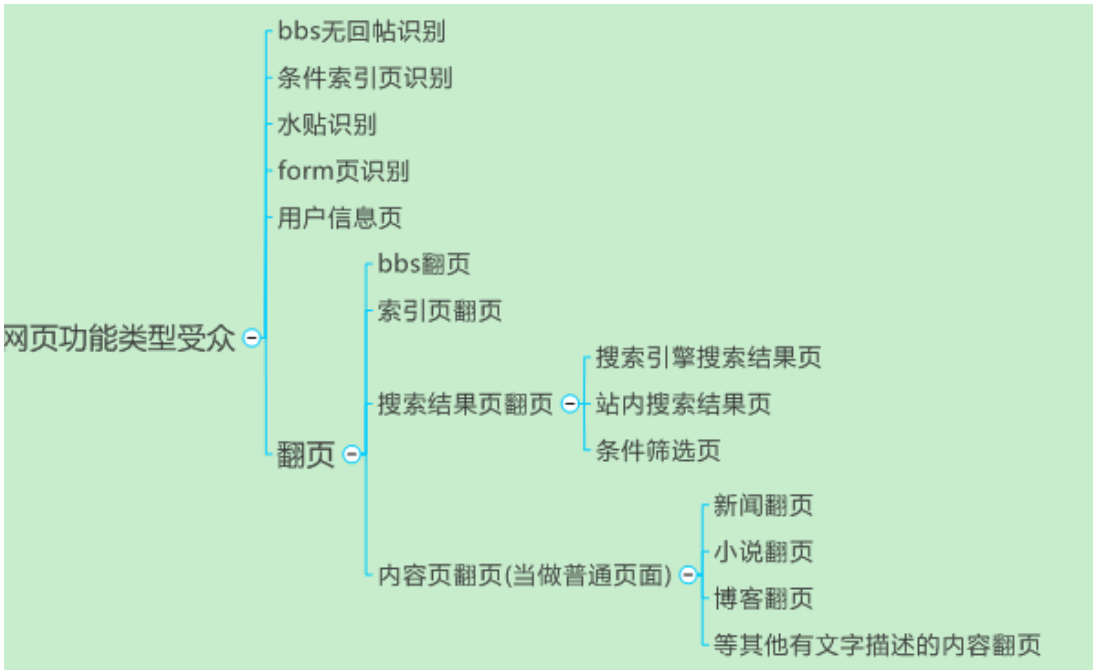

图 5-8 网页功能类型受众子维度

(2)源受众

源受众衡量页面前链、所在站点、所在集合（版面、用户、频道、etc）可满足需求的需求人群大小。目前该过滤系统仅考虑纳入一些已有的源受众特征，待受众维度的特征完善后再细化和扩展。如图 5-9 所示。

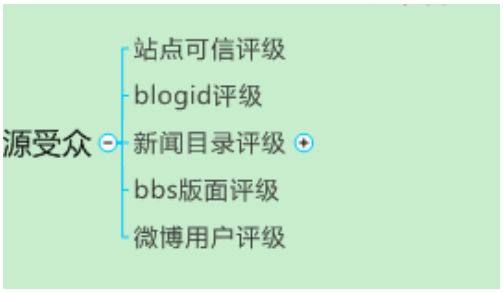

图 5-9 源受众子维度

5.2.2 非功能性需求

1）系统的可靠性：

该过滤系统属于 EC（计算中心）中的子系统，是对 HTML 网页包的实时打分计算的系统，要求该过滤系统能 7 X 24 小时连续运行，能够快速部署，因系统漏洞出现故障后能够及时快速的切换到备用机。

2）模型的可扩展性：

因为能够提供页面价值的因素非常多，所以子策略的个数会越来越多，所以多个策略的分数合并方法是非常重要的。而且每个策略自身的改进应该是独立的，不必调整其他的策略，即低耦合性。

3）系统的组件化：

即过滤系统本身可以单独使用，也可以与外部模块配合产生更好的结果。比如由页面分析传入一些子方面的结果（比如图片分档）以弥补过滤系统内部策略的不足，再如站点价值这样的数据可以传入，也可以自身装载词典。

5.2.3 系统架构

为了满足过滤系统的功能性和非功能性需求，我们首先对系统进行了模块划分逐一解决本章第一节和第二节提出的功能性需求和非功能性需求，其中包括解决功能性需求中的特征向量计算的网页价值计算模块和垃圾网页过滤模块；解决非功能性需求中模型的可扩展性需求的网页价值计算模型管理模块；由于系统的实际上线运作是运行与搜索引擎 spider 系统中的 EC 模块下作为计算组件（子系统），所以符合该过滤系统的系统组件化需求；在系统的可靠性方面，实际系统运作时将会有极大数量的服务器运行相同的

系统对服务进行相应，单个机器出现故障仅会在运行速度（执行效率）上有所减缓，并不会影响搜索引擎整体的运行。

图 5-10 体现了系统的模块架构。

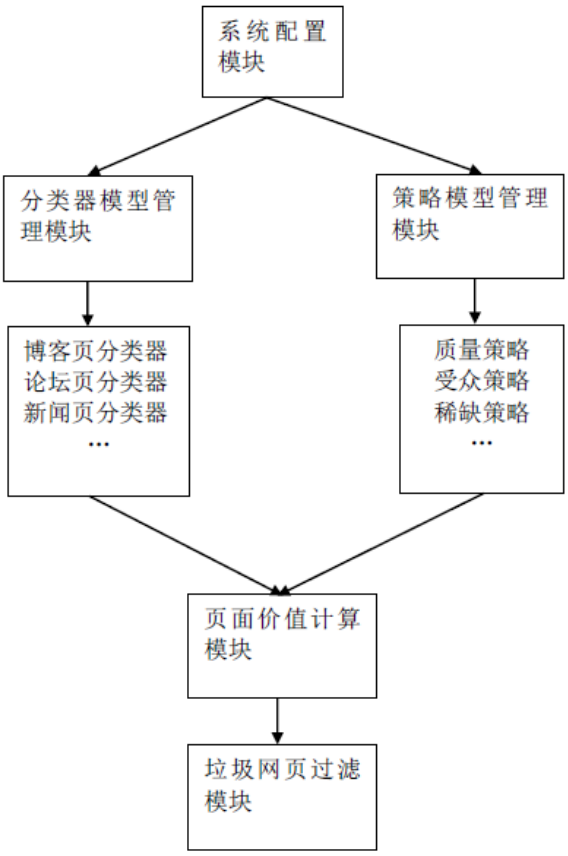

图 5-10 系统模块架构图

5.2.4 类设计

当该过滤系统开始运行时，首先由 PageValueConfig 类型的单实例读取相关配置，然后由 PageValueMachine 类型实例调用单实例工厂 AspectFactory 创建所有现存的(已开发完成并且已上线运行的模型）aspect 的实例；对于每一个 HTML 网页，需要依次通过所有的 aspect 进行特征向量的计算（图中展现的 Aspect 是该系统现有的子维度策略模型），最终由 ValueResult 整合各维度的特征值，通过网页价值计算模块进行最终整合打分得出该 HTML 网页的 score。其中，网页的基础特征保存在 FeatureSet 类中，这个类实现了按名字读取，可以直接根据字符串获得网页对应的属性。详细的类图设计如下图 5-11 所示。

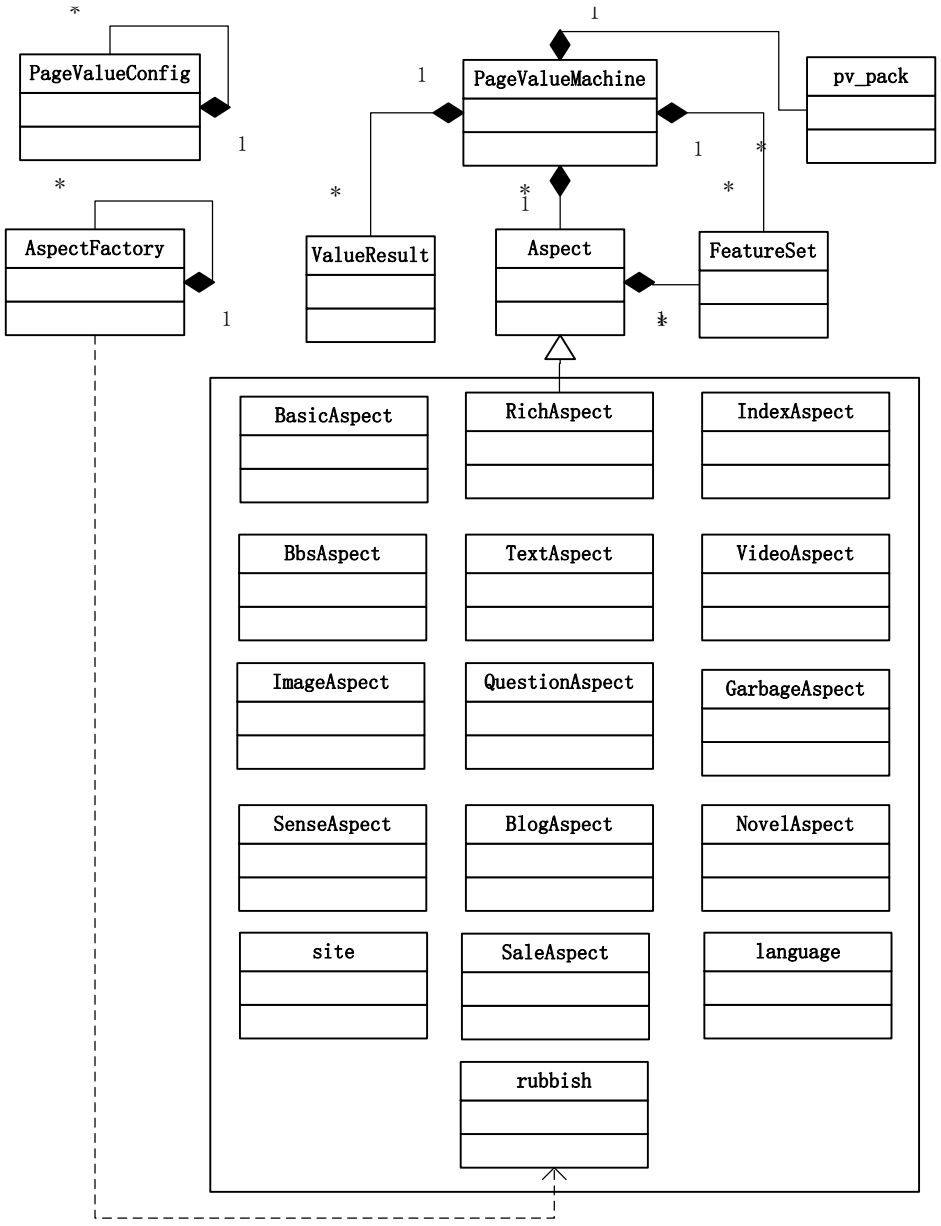

图 5-11 过滤系统类设计图

5.2.5 系统整体流程

该过滤系统是对传入进系统的 HTML 网页包进行多维度的打分计算。输入系统的网页包需要经过多个子维度下不同的策略模型进行计算，并将计算结果存储在 HTML 包头中。图 5-3 表现了该过滤系统的主要计算流程。

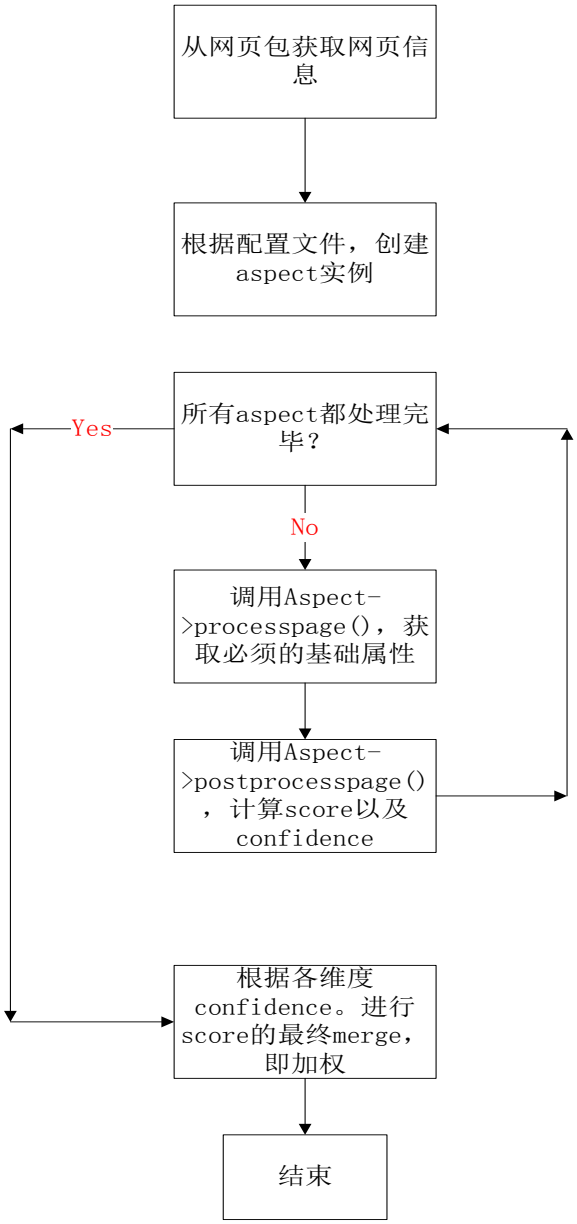

图 5-12 过滤系统主要计算流程图

5.3 详细设计与实现

本章将讨论从该过滤系统的各方面设计来描述整个系统的详细设计和实现。主要从以下几个模块设计开始讨论：网页分类器模型管理模块设计、网页价值计算模型管理模块设计、价值计算模块设计等，并给出该过滤系统中已有的网页分类器模型的开发实例。

5.3.1 分类器模型管理模块

过滤系统在第一次初始化时建立所有的策略对象，并将这些对象按 key 存储到一个 map 中，然后从配置文件中加载各种语言的策略路径，语言到策略路径的映射采用 std::map 结构，策略路径存储采用顺序容器以保证策略前后的依赖关系。

当传入进来的 HTML 网页包进入该过滤系统计算时，策略计算伪码大致如下：

```
Strategy_path strategy_list = strategy_path_map[language];
For each(strategy in strategy_list)
Begin
    Strategy->processpage(pagepack)
End
    Calculate pagevalue by result.
```

其中，*strategy_path_map* 是用来存储网页分类器模型映射的 map，从伪代码中可看出，过滤系统对网页的打分计算是先对所有的网页特征进行计算，后再整体综合的对该 HTML 网页进行按权重打分。通过这种方式管理策略，开发者能够非常便捷的对策略模型进行更改，满足需求分析中所要求的策略可扩展性。

图 5-13 是分类器管理模块中的开发者用例图。

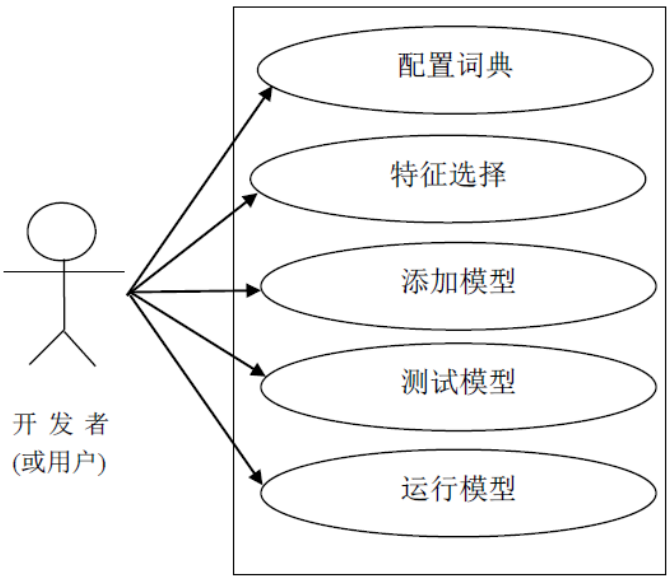

图 5-13 分类器管理模块下开发者用例图

开发者（或用户）可在分类器模型管理模块下进行配置词典、选择特征、添加模型、模型测试以及模型运行等主要功能。

5.3.2 策略模型管理模块

网页价值计算模型管理模块负责在本章第三节（5.3 节）中所描述的质量维度、受众维度等具体的特征的计算和子维度的打分，是整个过滤系统的基石。

网页价值计算模型管理模块基类如下:

```
Class Strategy
{
    public:
        Strategy (const char* name);
        virtual ~ Strategy ()=0;
    public:
        char name[MAX_ STRATEGY_NAME_LEN];
    public:
        virtual bool initlize();
        virtual bool destroy();
        virtual bool processPage(FeatureSet* pageinfo);
        virtual bool subValue(FeatureSet* pageinfo);
        virtual void addMiningInfo();
        virtual void dumpMiningInfo();
        virtual void clear();
    protected:
        FeatureSet* pageinfo_;
      char miningInfo_[MAX_MINING_INFO_LEN];
      comcfg::Configure * configure_;
};
```

其中 Strategy（策略）即为过滤系统中的各类型策略包括本文第二、三章所介绍的基于人工策略的网页分类器模型和基于机器学习的网页分类器模型以及该过滤系统中对 HTML 网页的特征计算和各维度打分的策略模型等。

下图是该过滤系统整体运行原理结构图，其中页面内部价值中即描述了网页价值计算管理模块。

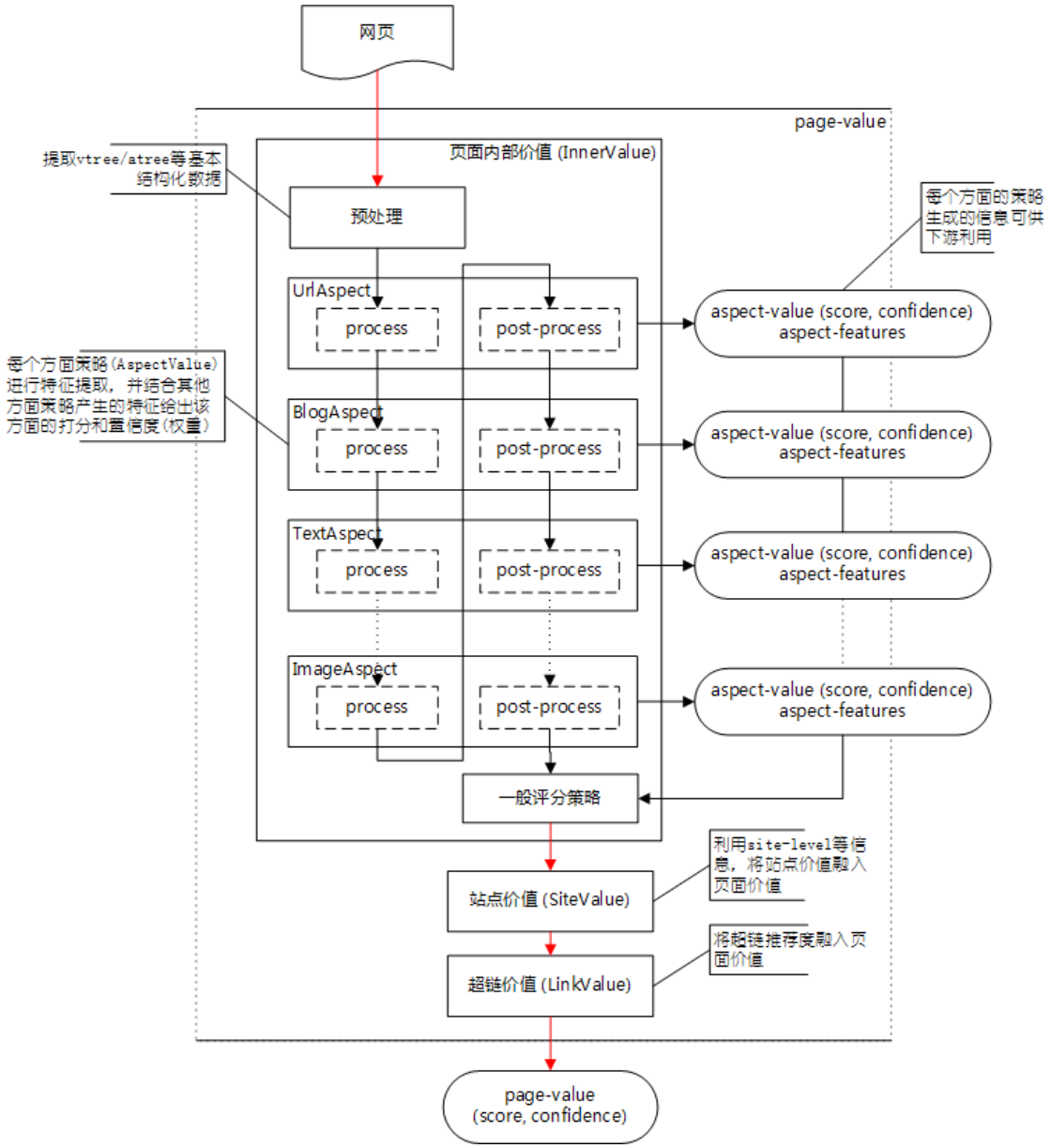

图 5-14 过滤系统整体运行原理结构图

如图所示每个子策略(aspect)基本上采用先提取基本特征，再综合各特征进行打分的方法，因为子策略之间可能要互相利用信息，因此每个子策略都分为两个处理阶段，在第一阶段完成可独立计算的部分，在第二阶段根据其他子策略的信息综合调整。

所有的特征都放入一个叫做 PageInfo 的结构中，以便各子策略之间传递特征，同时也供外部集中获取页面全部特征之用。

“一般评分策略”是指综合各子策略提取的特征，采用机器学习等手段得到一个一般性的分数，以弥补各策略中人工逻辑的不足。但目前还没有实现，只是给出了一个默认的常数分值。

图 5-15 是策略模型管理模块中的开发者（或用户）用例图。

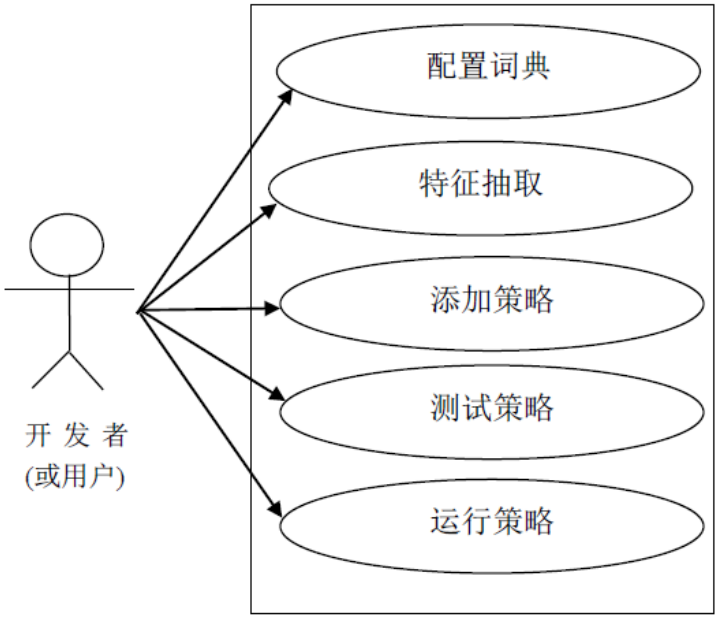

图 5-15 策略模型管理模块下开发者用例图

同分类器管理模块类似的，开发者（或用户）可在分类器模型管理模块下进行配置词典、特征抽取、添加模型、模型测试以及模型运行等主要功能。

5.3.3 页面价值计算模块

目前该过滤系统所设计使用的计算方式是在对一个网页执行完所有的策略之后，通过赋权计算，先汇聚到三维各自的打分，然后再根据赋权计算得出最终的打分，除此之外，还要实现一种快速生效机制，比如按某一特征或某一种特征组合进行直接打压或者提权豁免。所以最后的权值计算也设计成和价值策略的策略架构类似的方式来组织，避免出现代码零乱的现象。

以下公式为该过滤系统对页面内部价值打分的计算方法：

$$\text{页面内部价值分数} = \frac{\sum_i \text{子方面分数}_i \times \text{子方面置信度}_i + \text{一般分数} \times \text{一般置信度}}{\sum_i \text{子方面置信度}_i + \text{一般置信度}} \quad (5\text{-}1)$$

$$\text{页面内部价值置信度} = \min\left\{1, \sum_i \text{子方面置信度}_i + \text{一般置信度}\right\} \quad (5\text{-}2)$$

计算结构的设计思路：

1)优于各策略打分然后相加的方式。因为后者可能会各方面都低分但加起来比较高，或者几方面的低抵消了关键方面的高。

2）优于各策略打分然后固定权重的相加的方式。因为各个方面的权重对不同的页面应该是不一样的。

3）优于人工决策树的方式。后者当策略逐渐增多的时候不易扩展。

为了在分数合并之后得到一个合理的总体打分，每个策略的制定有如下指导原则：

1）对于资源型片面策略（比如文本、图片、视频），有资源的页面高分高权重。因为从这一片面来讲就可以决定页面为高价值页面，无论其他片面是否有价值。

2）对于资源型片面策略，无资源的页面低分低权重。因为这一方面资源的缺失完全不能说明这个页面无价值，只有所有片面的价值都低时，页面整体才会低价值。

3）对于作弊页面，低分高权重，因为从作弊就可以判定页面低价值（对于色情页面还不一定）。 非资源型片面策略，比如搜索结果页、商品页等等，也可能给出低分高权重。

5.3.4 垃圾网页过滤模块

过滤系统本身是一个页面信息挖掘的子系统，之所以称之为过滤系统，正是因为目前该系统线上的实际作用就是过滤垃圾网页，减少垃圾网页、低质网页进入网页库，占用大量存储空间。对于过滤功能来说，用户则可看成是搜索引擎建库方，为了提升网页库的有效性，便可以使用该过滤模块进行网页库的实时过滤，或是对现有的网页库进行“刷库”，也就是对库内所有网页进行打分计算，将被识别为垃圾的网页从网页库中删去。

该模块的实现主要是通过之前的页面价值计算模块对 HTML 包计算得出的结果后，再进行筛选。当用户需要过滤的是垃圾网页的时候，便在该模块设置过滤规则，如 score 为 0 的网页（即垃圾网页）；当用户需要不仅过滤垃圾网页，还需要过滤低质网页的时候，同样更改过滤规则将 score 小等于 1 的网页（即垃圾网页和低质网页）即可。

用户用例图可参考图 5-15.

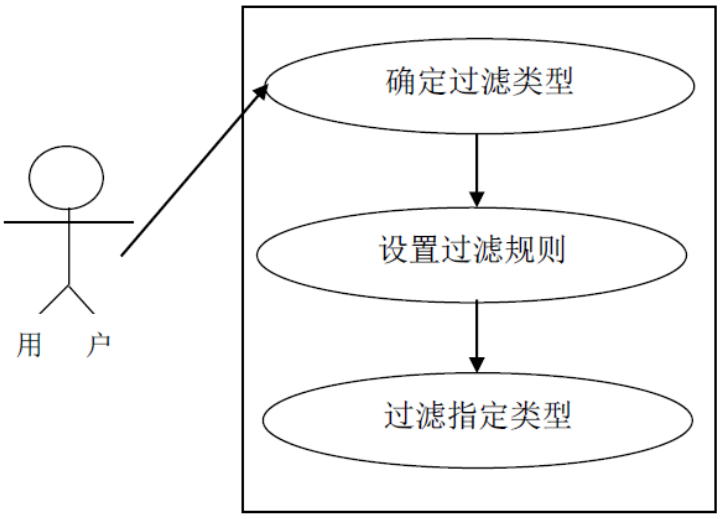

图 5-16 垃圾网页过滤模块用例图

5.4 系统测试

过滤系统的测试主要从测试过程和测试方法两方面描述。

5.4.1 测试流程

该过滤系统的测试流程主要分两方面对过滤系统进行测试：

1）对各模块进行整体的测试

（1）页面信息准确性测试：测试页面信息（特征抽取）是否准确。

（2）子维度打分准确性测试：测试各个子维度（aspect）的计算（特征计算）是否准确。

（3）分数整合的正确性：主要测试页面价值计算模块中分数加权的方式是否正确。

2）主要测试策略模型、分类器模型，分数与基础属性相关；主要通过构造基础属性结构体来进行检验；无法在系统中直接构造并运行测试的子维度，可以通过网页评分标准进行抽样调查。

5.4.2 测试方法

1) 从日志中抽取各个子维度计算后的基础属性，并与人为实际计算的属性进行对比，从而验证过滤系统中页面价值计算模块计算的基础属性的正确性；

2) 构造基础属性结构，验证分数是否符合预期，这部分比较少，由于大多数打分策略比较直观，通过得出数据直接观察就可以解决问题，不需要自动化进行解决；

3) 各维度分数进行汇总整合后的正确性：这部分的测试主要是一个加权算法，通过构造数据并运行测试即可；

结　论

作者在整个公司小组的带领下和多位公司同事的在多方面的耐心且细心的指导下，在近半年的实习期间经过不断努力成功完成了公司实际项目的开发任务，且所开发的网页分类器模型取得了超出开发前期所调研预期的效果。

本文详细描述了项目所采用的开发网页分类器模型的开发模式及模型结构和模型所应用的技术及技术原理等，且在本文的最后章节还对该项目开发环境所在的搜索引擎中用以给网页进行多方面维度打分计算的过滤系统进行了从系统设计背景、系统功能需求、整体框架、各模块的详细设计和实现等方面的详细描述。

在实际项目当中，作者通过调研实验和实际模型的开发，完善了基于人工策略模型开发的开发模式，并且通过对基于机器学习算法的网页分类器模型的调研实验，比较了不同的机器学习算法的效果，并得出了实验数据，对机器学习模型的开发提供了数据支持等。且在实习期间历时三个月所开发的作弊页识别模型于年初正式上线，且如今实时运行并帮助过滤系统识别过滤进入网页库的垃圾网页。

实习期间项目开发的过程也是作者不断学习和成长的过程。在这段时间里，作者遇到问题时候向项目团队的同事学习请教，认真查阅书籍或上网寻找答案，自身的技能和专业水平有了大的提高。但自身的不足以及以后还要继续努力学习的方面是无止尽的，尤其通过本次论文的撰写，也感受到了自身存在的差距，需要继续努力，弥补自身的不足。

参考文献

[1] 张磊. 《虚拟社区不良信息过滤技术研究》[硕士论文]. 昆明理工大学. 2011

[2] 屈军. 《基于增量的贝叶斯算法在网页文本中的应用》. 赤峰学院学报(自然科学版). 2013

[3] Tom Mitchell, McGraw Hill, Machine Learning, 1997.

[4] 杨芹. 《基于最大熵模型的中文网页分类器设计和实现》[硕士论文]. 苏州大学. 2010

[5] Daniel Riboni, Feature Selection for Web Page Classification, D.S.I, University’s degli Studi di Milano, Italy

[6] 姚建民. 《半结构化网页中商品属性抽取方法研究》[硕士论文]. 苏州大学. 2013

[7] XIAOGUANG QI, BRIAN D. DAVISON, Web Page Classification: Features and Algorithms

[8] 段军峰，黄维通，陆玉昌 《中文网页分类研究与系统实现》. 计算机科学 2007V01.34No.6

[9] 焦莉娟. 《基于最大熵模型的网页分类》[硕士论文]. 山西大学. 2006

[10] 王伟. 《基于语义挖掘的智能竞争情报系统研究》情报理论与实践. 2008

[11] 李晋松. 《基于朴素贝叶斯的网页自动分类技术研究》[硕士论文]. 北京化工大学. 2008

[12] 李玲俐. 《数据挖掘中分类算法综述》. 重庆师范大学学报(自然科学版). 2011

[13] 中国科学院研究生院（计算技术研究所）博士论文 李素建-《汉语组块计算的若干研究》-2002

[14] 郭宏斌. 《我国重点新闻网站发展现状及对策研究》[硕士论文]. 中南大学. 2011

[15] 冯兴华.《基于公理模糊集的模糊决策树算法研究》[博士论文]. 大连理工大学. 2013

[16] 郑敏. 《网络教育领域中文网页分类表的编制及应用研究》[硕士论文]. 中山大学. 2004

[17] 薛永大. 《网页分类技术研究综述》. 电脑知识与技术. 2012 年 25 期

[18] 姚建民. 《半结构化网页中商品属性抽取方法研究》[硕士论文]. 苏州大学. 2013

[19] 李灵华 米守防. 《国外典型元搜索引擎特性比较与分析》. 计算机工程与设. 2010 年 9 期

[20] 肖雪. 《中文文本层次分类研究及其在唐诗分类中的应用》[硕士论文] 重庆大学. 2006

[21] 刘志伟. 《基于组合优化的线性分类算法研究》[硕士论文] 西安电子科技大学. 2013

[22] 闫晓飞 陈良臣 孙功星. 《支持向量机多类分类算法的研究》第 13 届全国计算机、网络在现代科学技术领域的应用学术会议. 2007-11-04

[23] 徐久成 刘洋洋 杜丽娜 孙林. 《基于三支决策的支持向量机增量学习方法》. 计算机科学. 2015 年 6 期

[24] 张婷婷. 《基于 ARMA 模型的时间序列挖掘》[硕士论文]. 2013

[25] 李金 周璐璐 于虹 梁洪. 《基于支持向量机的体数据分类算法研究》. 系统仿真学报 2009

致　　谢

作者在编写本论文期间是在导师 XXX 老师的悉心指导下完成的，从论文最初的选题、改题、框架设计到论文的正文的编写、后期的修改、论文规范等都得到了 XXX 老师的细心指导和坚定支持。在实习期间至毕业论文的编写过程，XXX 老师丰富的实践经验、深厚的理论知识、严谨的科研态度让作者受益匪浅。没有 XXX 老师的极力帮助，作者恐难顺利完成毕业设计论文的编写，在此，谨向 XXX 老师表示衷心的感谢。

此外，作者还得到了公司同事和项目小组成员的热情帮助和极力指导。在作者的实习期间，公司同事在工作、学习、熟悉企业文化等等方面都给予作者了极大的帮助和指导，在此也向公司同事们表示衷心感谢。

在论文的编写期间，还得到了来自宿舍同学和毕设小组成员的帮助，不仅在论文编写架构设计上，还有论文及毕设材料的准备提交上也得到他们的极力帮助，在此同样感谢他们对作者的帮助。

本文虽已完成，但毕竟本科阶段学识尚浅，论文中定有不少浅显不足之处，望请批评指正，谢谢。

附 录 A 英文文献

Web Page Classification: Features and Algorithms

XIAOGUANG QI and BRIAN D. DAVISON

Lehigh University

Classification ofWeb page content is essential to many tasks inWeb information retrieval such asmaintaining Web directories and focused crawling. The uncontrolled nature ofWeb content presents additional challenges to Web page classification as compared to traditional text classification, but the interconnected nature of hypertext also provides features that can assist the process. As we review work in Web page classification, we note the importance of these Web-specific features and algorithms, describe state-of-the-art practices, and track the underlying assumptions behind the use of information from neighboring pages.

1. INTRODUCTION

Classification plays a vital role in many information management and retrieval tasks. On the Web, classification of page content is essential to focused crawling, to the assisted development of web directories, to topic-specific Web link analysis, to contextual advertising, and to analysis of the topical structure of the Web. Web page classification can also help improve the quality of Web search. In this survey we examine the space of Web classification approaches to find new areas for research, as well as to collect the latest practices to inform future classifier implementations. Surveys in Web page classification typically lack a detailed discussion of the utilization of Web-specific features. In this survey, we carefully review the Web-specific features and algorithms that have been explored and found to be useful for Web page classification. The contributions of this survey are

—a detailed review of useful Web-specific features for classification;
—an enumeration of the major applications for Web classification; and
—a discussion of future research directions.

The rest of this article is organized as follows: the background of Web classification and related work are introduced in Section 2; features and algorithms used in classification are reviewed in Sections 3 and 4, respectively; we discuss several related issues in Section 5, and point out some interesting directions and conclude the article in Section 6.

2. BACKGROUND AND RELATED WORK

Before reviewing Web classification research, we first introduce the problem, motivate it with applications, and consider related surveys in Web classification.

2.1. Problem Definition

Web page classification, also known as*Web page categorization*, is the process of assigning aWeb page to one or more predefined category labels. Classification is traditionally posed as a supervised learning problem [Mitchell 1997] in which a set of labeled data is used to train a classifier which can be applied to label future examples. The general problem ofWeb page classification can be divided into more specific problems: subject classification, functional classification, sentiment classification, and other types of classification. Subject classification is concerned about the subject or topic of a Web page. For example, judging whether a page is about “arts,” “business,” or “sports” is an instance of subject classification. Functional classification cares about the role that the Web page plays. For example, deciding a page to be a “personal homepage”, “course page” or “admission page” is an instance of functional classification. Sentiment classification focuses on the opinion that is presented in a Web page, that is, the author’s attitude about some particular topic. Other types of classification include genre

classification (e.g., zu Eissen and Stein [2004]), search engine spam classification (e.g., Gy ̈ongyi and Garcia-Molina [2005b]; Castillo et al. [2007]), and so on. This survey focuses on subject and functional classification. Based on the number of classes in the problem, classification can be divided into binary classification and multiclass classification, where binary classification categorizes instances into exactly one of two classes (as in Figure 1(a)), and multiclass classification deals with more than two classes. Based on the number of classes that can be assigned to an instance, classification can be divided into single-label classification and multilabel classification. In single-label classification, one and only one

class label is to be assigned to each instance, while in multilabel classification, more than one class can be assigned to an instance. If a problem is multiclass, for example, four-class classification, it means four classes are involved, for example, Arts, Business, Computers, and Sports. It can be either single-label, where exactly one class label can be assigned to an instance (as in Figure 1(b)), or multilabel, where an instance can belong to any one, two, or all of the classes (as in Figure 1(c)). Based on the type of class assignment, classification can be divided into hard classification and soft classification. In hard classification, an instance can either be or not be in a particular class, without an intermediate state, while in soft classification, an instance can be predicted to be in some class with some likelihood (often a probability distribution across all classes, as in Figure 1(d)).

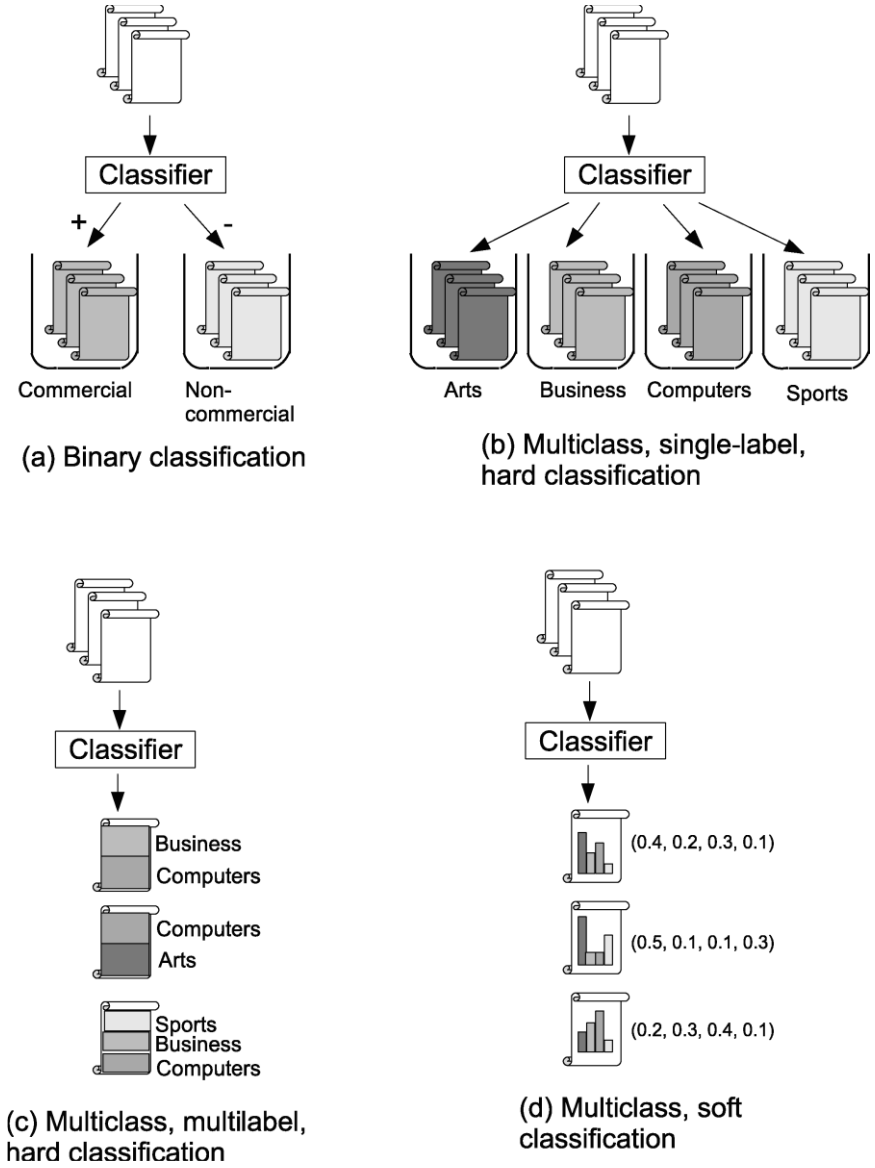

Fig. 1. Types of classification.

Based on the organization of categories, Web page classification can also be divided into flat classification and hierarchical classification. In flat classification, categories are considered parallel, that is, one category does not supersede another, while in hierarchical classification, the categories are organized in a hierarchical tree-like structure, in which each category may have a number of subcategories. An illustration is shown in Figure 2. Section 4 will address the issue of hierarchical classification further.

2.2. Applications of Web Classification

As briefly introduced in Section 1, classification of Web content is essential to many information retrieval tasks. Here, we present a number of such tasks.

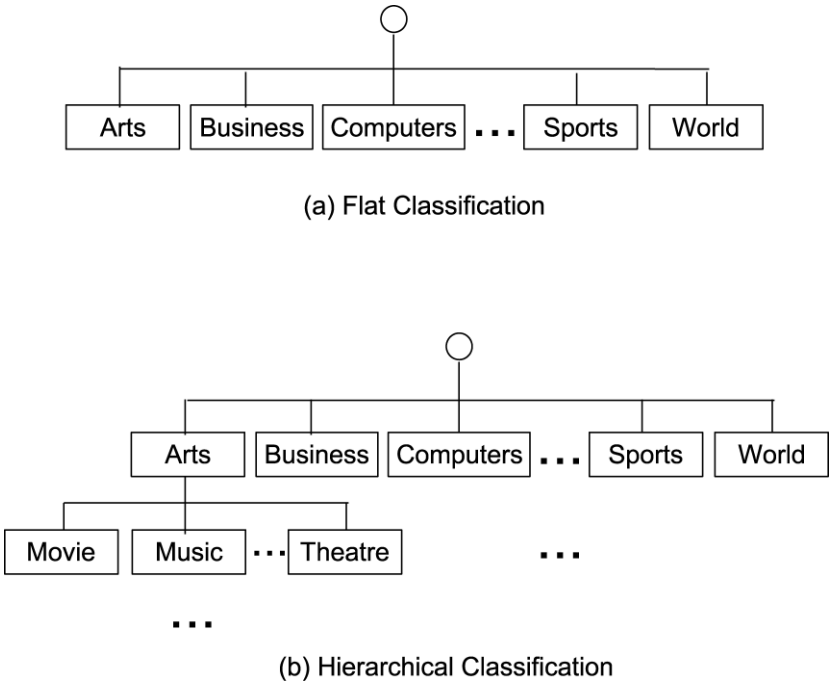

Fig. 2. Flat classification and hierarchical classification

2.2.1. Constructing, Maintaining or Expanding Web Directories (Web Hierarchies). Web directories, such as those provided by Yahoo!1 and the dmoz Open Directory Project (ODP),2 provide an efficient way to browse for information within a predefined set of categories. Currently, these directories are mainly constructed and maintained by editors, requiring extensive human effort. As of February 2008, it was reported [Netscape Communications Corporation 2008] that there were 78,940 editors involved in the dmoz ODP. As the Web changes and continues to grow, this manual approach will become less effective. One could easily imagine building classifiers to help update and expand such directories. For example, Huang et al. [2004a, 2004b] proposed an approach for the automatic

creation of classifiers from Web corpora based on user-defined hierarchies. Furthermore, with advanced classification techniques, customized (or even dynamic) views of Web directories can be generated automatically. There appears to be room for further interesting work on automatic construction of Web directories.

2.2.2. Improving Quality of Search Results.

Query ambiguity is among the problems that undermine the quality of search results. For example, the query term *bank* could mean the border of a body of water or a financial establishment. Various approaches have been proposed to improve retrieval quality by disambiguating query terms. Chekuri et al. [1997] studied automatic Web page classification in order to increase the precision ofWeb search. A statistical classifier, trained on existingWeb directories, is applied to new Web pages and produces an ordered list of categories in which the Web page can be placed. At query time the user is asked to specify one or more desired categories so that only the results in those categories are returned, or the search engine returns a list of categories under which the pages would fall. This approach works when the user is looking for a known item. In such a case, it is not difficult to specify certain about what documents will match, for which the above approach does not help much. Search results are usually presented in a ranked list. However, presenting categorized, or clustered, results can be more useful to users. An approach proposed by Chen and Dumais [2000] classifies search results into a predefined hierarchical structure and presents the categorized view of the results to the user. Their user study demonstrated that the category interface is liked by the users better than the result list interface, and is more efficient for users to find the desired information. Compared to the approach suggested by Chekuri et al. [1997], this approach is less efficient at query time because it categorizes Web pages on-the-fly. However, it does not require the user to specify desired categories; therefore, it is more helpful when the user does not know the query terms well. Similarly, K ¨aki [2005] also proposed presenting a categorized view of search results to users. Experiments showed that the categorized view is beneficial for the users, especially when the traditional ranking of results is not satisfactory. Page et al. [1998] developed the link-based ranking algorithm called *PageRank*. PageRank calculates the authoritativeness of Web pages based on a graph constructed by Web pages and their hyperlinks, without considering the topic of each page. Since then, research has been conducted to differentiate authorities of different topics. Haveliwala [2003] proposed Topic-Sensitive PageRank, which performs multiple PageRank calculations, one for each topic. When computing the PageRank score for each category, the

random surfer jumps to a page in that category at random rather than just any Web page. This has the effect of biasing the PageRank to that topic. This approach needs a set of pages that are accurately classified. Nie et al. [2006] proposed anotherWeb-ranking algorithm that considers the topics ofWeb pages. In that method, the contribution that each category has to the authority of a Web page is distinguishe by means of soft classification, in which a probability distribution is given for a Web page being in each category. In order to answer the question “To what granularity of topic the computation of biased page ranks make sense?” Kohlschutter et al. [2007] conducted an analysis on ODP categories, and showed that ranking performance increases with the ODP level up to a certain point. It seems further research along this direction is quite promising.

2.2.3. Helping Question Answering Systems. A question-answering system may use classification

techniques to improve its quality of answers. Yang and Chua [2004a, 2004b] suggested finding answers to list questions (where a set of distinct entities are expected, e.g., “Name all the countries in Europe”) through a Web page functional classification. Given a list question, a number of queries are formulated and sent to search engines. TheWeb pages in the results are retrieved and then classified by decision tree classifiers into one of the four categories: collection pages (containing a list of items), topic pages (representing an answer instance), relevant pages (supporting an answer instance), and irrelevant pages. In order to increase coverage, more topic pages are included by following the outgoing links of the collection pages. After that, topic pages are clustered, from which answers are extracted. There have additionally been a number of approaches to improving the quality of answers by means of question classification [Harabagiu et al. 2000; Hermjakob 2001; Kwok et al. 2001; Zhang and Lee 2003] which are beyond the scope of this survey. One interesting question that previous publications have not answered is how useful Web page subject classification is in question answering systems. In Section 2.2.2, we reviewed a number of approaches that use the topical information of Web pages to improve the performance of Web search. Similarly, by determining the category of expected answers to a question and classifying the Web pages that may contain candidate answers, a question answering system could benefit in terms of both accuracy and efficiency.

2.2.4. Building Efficient Focused Crawlers or Vertical (Domain-Specific) Search Engines.

When only domain-specific queries are expected, performing a full crawl is usually inefficient. Chakrabarti et al. [1999] proposed an approach called *focused crawling*, in which only documents relevant to a predefined set of topics are of interest. In this approach, a classifier is used to evaluate the relevance of a Web page to the given topics so as to provide evidence for the crawl boundary.

2.2.5. Other Applications.

Besides the applications discussed above, Web page classification is also useful in Web content filtering [Hammami et al. 2003; Chen et al. 2006], assisted Web browsing [Armstrong et al. 1995; Pazzani et al. 1996; Joachims et al. 1997], contextual advertising [Broder et al. 2007a, 2007b], ontology annotation [Seki and Mostafa 2005], and knowledge base construction [Craven et al. 1998].

2.3. The Difference Between Web Classification and Text Classification

The more general problem of text classification [Sebastiani 1999, 2002; Aas and Eikvil 1999; Tan 1999; Tong and Koller 2001; Cardoso-Cachopo and Oliveira 2003; Bennett et al. 2005] is beyond the scope of this article. Compared with standard text classification, classification ofWeb content is different in the following aspects. First, traditional text classification is typically performed on structured documents written with consistent styles (e.g., news articles) [Chekuri et al. 1997], while Web collections do not have such a property. Second, Web pages are semistructured documents in HTML, so that they may be rendered visually for users. Although other document collections may have embedded information for rendering and/or a semistructured format, such markup is typically stripped for classification purposes. Finally, Web documents exist within a hypertext, with connections to and from other documents. While not unique to the Web (consider e.g., the network of scholarly citations), this feature is central to the definition of the Web, and is not present in typical text classification problems. Therefore, Web classification is not only important, but distinguished from traditional text classification, and thus deserving of the focused review found in this article.

2.4. Related Surveys

Although there are surveys on textual classification that mention Web content, they lack an analysis of features specific to the Web. Sebastiani [2002] mainly focused on traditional textual classification. Chakrabarti [2000] and Kosala and Blockeel [2000] reviewedWeb mining research in general as opposed to concentrating on classification. Mladenic [1999] reviewed a number of text-learning intelligent agents, some of which areWeb-specific. However, her focus was on document representation and feature selection. Getoor and Diehl [2005] reviewed data mining techniques which explicitly consider links among objects, with Web classification being one of such areas. F ¨urnkranz [2005] reviewed various aspects of Web mining, including a brief discussion on the use of link structure to improve Web classification. Closer to the present article is the work by Choi and Yao [2005] which described the state-of-the art techniques and subsystems used to build automatic Web page classification systems. This survey updates and expands on prior work by consideringWeb-specific features and algorithms in Web page classification.

附　　录　B 中文翻译

网页分类：　特征和算法

XIAOGUANG QI and BRIAN D. DAVISON
Lehigh University

网页内容的分类是许多关于网络信息检索任务的关键，例如维护网站目录和重点抓取等任务。相比于传统的文本分类，网页内容的不可控性体现了对网页分类更大的挑战，但是超文本的互联性同时也提供了帮助实现这一进程的诸多特征。

正如我们反观网页分类的工作，我们发现这些网络的特征和算法的重要性，描述最前沿的实践和跟踪最先的关于相邻网页信息的利用的假想。

1. 简介

在许多信息管理和检索任务中，分类扮演了至关重要的角色。在互联网中，网页内容的分类对于信息抓取，网络索引的协助开发，特定话题的网页链接的分析和上下文分析，分析网站话题结构等都是非常重要的。

在本调查中，我们考察了 Web 分类方法的空间，找到了新的研究领域，收集了最新的做法，以便实现将来的分类技术。一般情况下，在网页分类调查中，都缺少详细关于具体的网络特征的应用讨论。在本次调查中，我们仔细的查看了现如今已有的且被公认为较为有用的网页类型分类的特征和算法。本调查的贡献如下。

——对于公认有用的网络分类特征的详细描述
——对 Web 分类主要应用的枚举
——未来的研究方向的讨论

本文的其余部分安排如下：网页分类的背景和相关的工作进行介绍（第二章节），在分类使用的功能和算法综述（第三四章节）我们（在第五章节）讨论了几个相关的问题，并指出一些有趣的方向，并总结整篇文章。

2. 背景及相关工作

回顾网络分类研究之前，我们先介绍一下这个问题，激励它的应用，并考虑网页分类的相关调查。

2.1 问题定义

网页分类，又称网页归类，是一个分类的过程，将网页分类到一个或多个预定类别的标签。分类是传统的作为一个监督的学习的问题[米切尔 1997]其中一组训练数据用于训练可应用到标注未来实例的分类器。

一般情况下的网页分类，可分为更具体的问题：主题分类，功能分类，情感分类，以及其他类型的分类。学科分类是关注的主题或主题网页。例如，判断是否一个网页是关于“艺术”,“商业”或“运动”是主题分类的一个实例。功能分类主要作用是， 例如在决定一个页面是一个“个人主页”,“课程页面”或“录取页”是功能分类的一个实例。情绪

分类侧重于在网页中提出意见，一般是作者的一些特定主题的态度。其他类型的分类包括流派分类（例如 zu Eissen and Stein [2004]），搜索引擎的垃圾邮件分类（例如 Gy¨ongyi and Garcia-Molina [2005b]; Castillo et al. [2007]）等。调查重点主体和功能分类。

基于在问题的类的数量，分类，可分为二进制分类和多分类，其中二元分类分类实例分成两类的正好一个（如在图 1（a）），以及多类分类涉及两个以上的分类。根据可分配的类的数量，分类可分为单标签分类和多标签分类。在单标签分类，且只有一个类标签是要分配到每一个实例，而在多标记分类，一个以上的类将分配给一个实例。如果一个问题是多类分类，例如，四类分类，这意味着四个分类都参与其中，例如，艺术，商业，计算机和体育。

它可以是单标签，其中，恰好一个类可以被分配到一个实例（如在图 1（b）），或者多标签，其中一个实例可以属于任何一个，两个或所有的类别（如在图 1 的（c））。基于类别指定的类型，分类可分为硬分类和软分类。在硬分类中，实例可以是或不是在一

个特定的类，没有中间状态，而在软分类中，一个实例可以被预测为在某些类的一些的可能性（通常在所有的类的概率分布，如在图 1（D））。

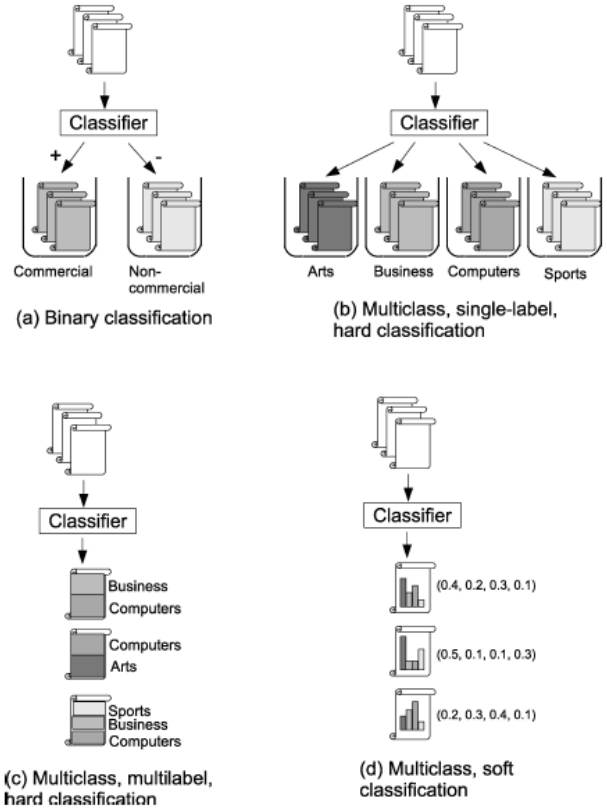

图 1. 分类的种类

2.2 网页分类的应用

如同第一节的简要介绍，网络内容分类是许多重要信息检索任务之一。这里，我们提出了许多这样的任务。

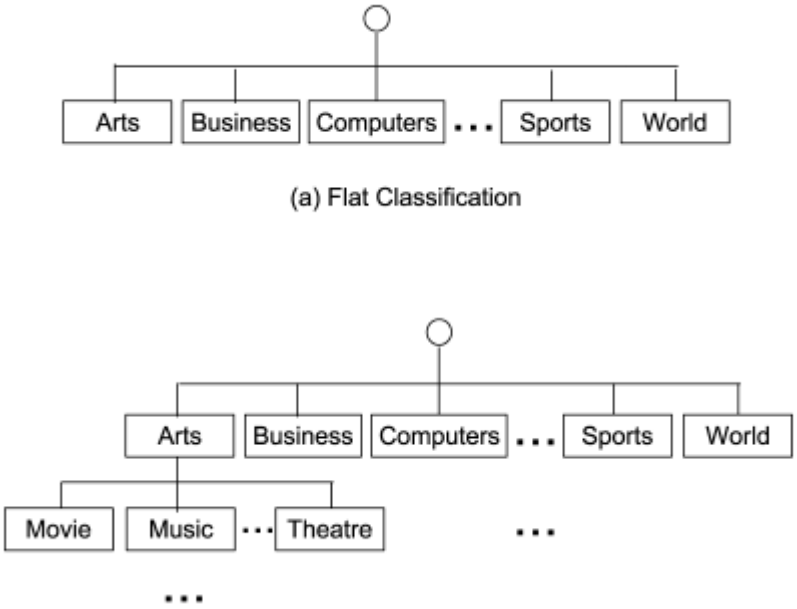

图 2 非层次分类和层次分类

2.2.1 构建，维护和扩展 Web 目录（网络层次结构）

网站目录，如雅虎 1 和 DMOZ 开放目录项目（ODP），2 提供了一种有效的方法来预定的一组类别中浏览信息。目前，这些目录主要是建造和维护编辑，要求大量的劳动力。截至 2008 年 2 月，据报道[网景通讯公司 2008]，有参与 DMOZ ODP 78940 编辑器。随着网络的变化和持续增长，这种手工方法将变得不那么有效。人们可以很容易想象构建分类，帮助更新和扩大这种类别。例如，Huang 等人。 [2004 年 a，2004 年 b]提出了一种方法基于网页语料分类的的自动生成基于用户定义的层次结构。 此外，先进的分类技术，静态（甚至动态）可以自动生成浏览网页的类别。甚至还可以基于 Web 目录自动构建更有趣的工作。

2.2.2 提高搜索质量

查询模糊是导致搜索质量差的主要问题。例如，查询短期银行可能意味着水体或金融机构的边界。现如今各种解决的方法大多提出了歧义查询条件来提高检索质量。Chekuri 等。 [1997]为了提高精度研究自动网页分类 ofWeb 搜索。统计分类，训练了现存网站目录，应用

到新的网页，并产生类的有序列表，其中该网页可以放置。在查询时，用户被要求指定一个或多个期望的类别，因此，只有在这些类别的结果被返回，或搜索引擎返回根据

该网页将属于类别的列表。这种方法适用于当用户正在寻找一个已知项。在这样的情况下，它并非难以指定类别的。然而，在有些情况下，用户不太确定什么文档与之匹配，为此，上述做法不利于许多情况。

搜索结果通常在排名列表中显示。然而，呈现归类，或簇生，结果可以给用户更加有用。陈提出的方法和杜迈斯[2000]搜索结果划分为一个预定义的层次结构和呈现结果的用户的分类的图。他们的用户研究表明该类别接口由用户比结果列表更友好的界面，并且更高效的使用户找到所需的信息。相比由 Chekuri 等人建议的方法。 [1997]，这种方法是在低效率

查询时间，因为它实时的归类网页。然而，它不需要用户指定所希望的类别;因此，它是在为用户执行更多的帮助，但不知道查询条件良好与否。同样，柿[2005]还提出呈现出分类

查看搜索结果的用户。实验表明，分类视图是对用户是有益的，特别是当结果的传统排名不满意的时候。

[1998]开发的基于链接的排名算法 PageRank。的 PageRank 计算构成 Web 页面的基于图形的权威性由网页和他们的超链接，不考虑每一个网页的主题。从那以后，研究已进行区分的不同主题。Haveliwala [2003]提出了主题敏感的 PageRank，它执行更多的 PageRank 计算，每个主题。在计算 PageRank 得分为每个类别中，随机的互联网用户随意跳转到页面该类别中，而不仅仅是任何网页。这有偏置的 PageRank 该主题的效果。这个方法需要一组被准确分类的网页。聂等人。 [2006]提出 anotherWeb 排名算法，考虑到主题网页页面。在该方法中，每个类别具有一个网页的权威的贡献是可分辨通过软分类的装置，在其中一个概率分布为 Web 定页面在各类别之中。为了回答这个问题:“要什么样的粒度话题偏向网页排名的计算更有意义？“Kohlschutter 等。 [2007]进行了 ODP 类别的分析，并表明排序的性能提升与 ODP 水平达到一定点。这似乎沿着这进一研究方向较有前景。

2.2.3 帮助答疑系统

一个问题与应答的系统可以使用分类技术，以改善其答案的质量。杨和 Chua[2004a，2004b]建议寻找答案列出的问题（其中一组不同的实体的预期，例如，通过网页功能分类“名称的所有欧洲国家”)。给定一个列表问题，一些查询结果被配制和发送给搜索引擎。在结果网页被检索，然后通过决策树分类划分成四类中的一个：集合页（含有项目的列表），主题页（代表回答实例），相关网页（支持一个答案实例），以及不相关的网页。为了增加覆盖范围，主题页是由以下包括收藏页面的导出链接。在此之后，最终答案由从被提取的答案中专题页面聚集。

有还得到了一些方法来提高质量，例如按问题分类[Harabagiu 2000; Hermjakob2001;2001;Zhang and Lee2003]，它超出了我们本次的这个范围调查。

还有一个有趣的问题，以前的出版物都没有回答网页主题分类的问答系统是多么有用。在第 2.2.2 节中，我们回顾了一些使用网页的主题信息的方法以提高网络搜索的性能。类似地，通过确定的类别预期答案的问题和分类的网页可能含有候选答案，一个问答系统可以在两种方面受益——准确度和效率。

2.2.4 建立有效的聚焦爬行或垂直（领域特定）搜索引擎

当只有特定域的查询预计,执行完全抓取网站通常是低效的。查克拉巴蒂等。[1999]提出所谓的主题爬行的方法，其中只有相关的一组预定义的主题的文档的兴趣。在这种方法中，分类器是用于网页的相关性评价为给定的主题，以便为抓取边界提供信息。

2.2.5. 其他应用

除了上面讨论的应用之外，网页分类也是 Web 内容过滤[哈马米等人 2003; Chen 等人。 2006 年]辅助网络浏览[Armstrong 等。 1995; Pazzani 等。 1996; Joachims 等。1997 年]，上下文广告[布罗德等人。 2007 年 a，2007 年 b]，本体注释[关和 2005 年穆斯塔法]和知识库建设[克雷文等人。 1998 年]。

2.3 页面分类和文本分类的区别

文本分类的更一般的问题[塞巴斯蒂 1999 年，2002 年; AA 和 Eikvil1999;谭 1999;桐和 2001 年科勒;卡多佐-Cachopo 和 2003 奥利维拉;贝内特等。2005]超出了本文的范围。与标准的文本分类相比，分类网页内容在以下几个方面有所不同：首先，传统的文本分类上是一致的书面结构化文档通常执行样式（例如，新闻文章）[Chekuri 等。 1997 年]，而网页收藏中不具有这样的特性。其次，网页是在 HTML 文档半结构化，以便它们可以被用户可视地呈现。尽管其他文档集可能已嵌入的信息用于呈现和/或半结构化的格式，这种标记通常是剥去分类的目的。最后，Web 文档存在的超文本中，以向和从其他文件的连接。虽然没有独有的网络（例如考虑，学术引文网络），这个功能中心到 Web 的定义，并且不存在于典型的文本分类问题。因此，网页分类不仅是重要的，且区别于传统的文本分类，因此值得在此发现的重点审查文章。

2.4. 研究现状

虽然有提到 Web 内容的文本分类调查，他们缺乏具体到 Web 功能的分析。塞巴斯蒂亚尼[2002]主要集中在传统的文本分类。查克拉巴蒂[2000]和的 Kosala 和 Blockeel [2000]一般审视的网页挖掘研究，而不是专注于分类。Mladenic [1999]回顾了一些文本的学习智能代理的，其中一些网页特异性。然而，她的重点是文档表示和特征选择。Getoor 和迪尔[2005]审查的数据挖掘技术，这明确地考虑对象之间的联系，通过 Web 分类是这样的地区之一。 °Furnkranz [2005]Web 挖掘的审查各个方面，包括对使用环节的简要讨论结构，以提高网页分类。[Choi 和姚明[2005]这说明国家的最先进的技术和子系统用于构建自动网页分类系统。

本次调查的更新和扩展了以前的工作，考虑特定网络功能在网页分类算法。