分类号： TP311.5 单位代码： 10335

密 级： 无 学 号： Z134325137



硕士学位论文



**中文论文题目 ： 文本挖掘在新闻情报系统中的应用**

**英文论文题目： The Application of Text Mining in**

**News Intelligence System**

申请人姓名： 裘 虬

指导教师： 陈 岭

合作导师：

专业学位类别： 工程硕士

专业学位领域： 软件工程

所在学院： 软件学院

**论文提交日期 年 月 日**

**学位论文书脊示例**

|  |
| --- |
| **题**  **目**  **作**  **者**  **姓**  **名**  **浙**  **江**  **大**  **学** |

**中文论文题目** （小二号仿宋体加黑）



**论文作者签名:**

**指导教师签名:**

论文评阅人1：

评阅人2：

评阅人3：

评阅人4：

评阅人5：

答辩委员会主席：

委员1：

委员2：

委员3：

委员4：

委员5：

答辩日期：

**英文论文题目**（16pt Time New Roman，Bold）



**Author’s signature:**

**Supervisor’s signature:**

Thesis reviewer 1：

Thesis reviewer 2：

Thesis reviewer 3：

Thesis reviewer 4：

Thesis reviewer 5：

Chair：

（Committee of oral defence）

Committeeman 1：

Committeeman 2：

Committeeman 3：

Committeeman 4：

Committeeman 5：

Date of oral defence：

浙江大学研究生学位论文独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 **浙江大学** 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名： 签字日期： 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解 **浙江大学** 有权保留并向国家有关部门或机构送交本论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 **浙江大学** 可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名： 导师签名：

签字日期： 年 月 日 签字日期： 年 月 日

摘要

**关键词：**　文本挖掘、文本分类、情感分析、新闻资讯系统

Abstract

**Key Words：**,

目录

[摘要 i](#_Toc416691377)

[Abstract ii](#_Toc416691378)

[图目录 III](#_Toc416691379)

[表目录 IV](#_Toc416691380)

[第1章 绪论 1](#_Toc416691381)

[1.1 引言 1](#_Toc416691382)

[1.2 课题背景 1](#_Toc416691383)

[1.3 课题意义 1](#_Toc416691384)

[1.4 本文研究重点及工作安排 2](#_Toc416691385)

[第2章 新闻资讯系统及文本挖掘研究现状 3](#_Toc416691386)

[2.1 国内外新闻资讯系统研究现状 3](#_Toc416691387)

[2.2 自动化文本分类研究现状 5](#_Toc416691388)

[2.3 情感分析研究现状 5](#_Toc416691389)

[第3章 文本挖掘技术基础 6](#_Toc416691390)

[3.1 文本分类技术研究 6](#_Toc416691391)

[3.1.1 文本分类任务 6](#_Toc416691392)

[3.1.2 文本分类过程 6](#_Toc416691393)

[3.1.3 文本结构化 6](#_Toc416691394)

[3.1.4 文本分类算法 12](#_Toc416691395)

[3.1.5 文本分类性能指标 14](#_Toc416691396)

[3.2 情分析技术研究 15](#_Toc416691397)

[3.2.1 情感分析种类 15](#_Toc416691398)

[3.2.2 情感分析流程 15](#_Toc416691399)

[3.2.3 情感分析算法 15](#_Toc416691400)

[3.3 本章小结 16](#_Toc416691401)

[第4章 新闻资讯系统架构及功能 17](#_Toc416691402)

[4.1 新闻资讯系统架构设计 17](#_Toc416691403)

[4.2 新闻资讯系统模块设计 18](#_Toc416691404)

[4.2.1 爬虫系统设计 18](#_Toc416691405)

[4.2.2 数据库设计 20](#_Toc416691406)

[4.2.3 API设计 22](#_Toc416691407)

[4.2.4 ETL及文本分类 24](#_Toc416691408)

[4.2.5 热点抽取及情感分析 27](#_Toc416691409)

[4.2.6 搜索 27](#_Toc416691410)

[4.2.7 前端页面 29](#_Toc416691411)

[4.2.8 个性化推荐 33](#_Toc416691412)

[4.3 文本分类和情感分析在系统中的位置 34](#_Toc416691413)

[第5章 文本分类及情感分析实验 35](#_Toc416691414)

[5.1 文本分类比较实验 35](#_Toc416691415)

[5.1.1 实验环境设计 35](#_Toc416691416)

[5.1.2 文本分类数据来源 35](#_Toc416691417)

[5.1.3 实验结果分析 36](#_Toc416691418)

[5.2 情感分析实验 38](#_Toc416691419)

[5.2.1 情感分析算法实现 38](#_Toc416691420)

[5.2.2 情感分析数据来源 38](#_Toc416691421)

[5.2.3 情感分析实验结果呈现 38](#_Toc416691422)

[第6章 总结及展望 39](#_Toc416691423)

[6.1 总结 39](#_Toc416691424)

[6.1.1 39](#_Toc416691425)

[6.2 工作不足及展望 39](#_Toc416691426)

[6.2.1 39](#_Toc416691427)

[参考文献 42](#_Toc416691428)

[作者简历 44](#_Toc416691429)

[致谢 45](#_Toc416691430)

图目录

[图 2.1DIGG官网 4](#_Toc416691484)

[图 2.2极客公园官网 4](#_Toc416691485)

[图 4.1 系统MVC架构图 17](#_Toc416691486)

[图 4.2 个性化新闻情报系统流程架构 18](#_Toc416691487)

[图 4.3 爬虫系统架构图 19](#_Toc416691488)

[图 4.4 爬虫任务列表流程 19](#_Toc416691489)

[图 4.5 数据库设计 21](#_Toc416691490)

[图 4.6 ETL流程图 25](#_Toc416691491)

[图 4.7 文本分类流程图 26](#_Toc416691492)

[图 4.8 搜索框架流程 28](#_Toc416691493)

[图 4.9 倒排索引建立流程图 28](#_Toc416691494)

[图 4.10 个性化新闻资讯系统首页 29](#_Toc416691495)

[图 4.11 热门话题展示 30](#_Toc416691496)

[图 4.12 首页搜索框 31](#_Toc416691497)

[图 4.13 搜索结果页面 32](#_Toc416691498)

[图 4.14 个性化配置页面 33](#_Toc416691499)

[图 5.1 维度递减 37](#_Toc416691500)

[图 5.2 维度变化趋势 37](#_Toc416691501)

表目录

[表 3.1 分词工具比较 8](#_Toc416691502)

[表 3.2 卡方检验混淆矩阵 10](#_Toc416691503)

[表 3.3 性能检测混淆矩阵 14](#_Toc416691504)

[表 4.1 Article表结构及字段 22](#_Toc416691505)

[表 4.2 API设计 23](#_Toc416691506)

[表 4.3 Article\_classified表结构 27](#_Toc416691507)

[表 4.4 个性化推荐权重明细 33](#_Toc416691508)

[表 5.1 类别特征维数 36](#_Toc416691509)

# 绪论

## 引言

现如今，随着互联网的不断普及，互联网上的信息呈爆炸性的增长。据《中国互联网报告》指出，截止2014年6月，我国网民规模已达6.32亿，普及率相比13年年底提升了1.1个百分点，已达到46.9%。其中网络新闻应用的使用人数达到了5.03亿，达到了79.6%。与此同时，中国互联网站点也达到了270万个，其中新闻类占到了10%，按照每天每个网站更新10篇新闻资讯的量来计算，一天增加的新闻数量就达到了270万篇，并且因为大型门户网站和论坛的更新量远大于每天10篇，所以实际的文章数量会远远的超过这个数字。

对这些信息进行处理，提炼出有价值的知识是互联网企业甚至是传统企业增加自身核心竞争力，把握企业发展方向的一项非常有价值的活动。现在互联网上已经出现了舆情监控和资讯推荐类的软件或服务。但是将两者整合在一起的应用还很少。

同时每个公司都有自己的专门关注市场动态的部门，尤其是互联网公司，更是对互联网上的动态格外注意。但因为互谅网信息的繁杂，反而会产生噪声，有价值的信息相对较少，而大量的资讯信息的涌出势必会增加该部门的人力投入，以甄别高价值的资讯信息。

## 课题背景

课题来源于正是本人实习项目组中正在开发的一款面向企业级用户的新闻资讯及舆情监控系统。该系统旨在解决上面描述的问题，通过对因特网上大量的新闻进行深度的挖掘，对不同的用户，提供个性化的资讯推荐及舆情分析等功能。

本人在其间主要研究并负责文本挖掘的相关算法及应用。在文中，将着重阐述本人在实际开发过程中所面对的两个应用方向：文本分类及情感分析。

## 课题意义

目前数据挖掘、文本挖掘在工业界非常热，越来越多的人从事这个行业。与此同时，如何将理论运用到实践中成为了当务之急。并不是完美的理论都能完美的运用到实践中。邓小平说过，实践是检验真理的唯一途径；不管是白猫黑猫，能抓到老鼠的就是好猫。

本课题的意义旨在针对应用层面上，对现如今流行的文本挖掘算法进行研究和学习，通过实验的方式挑选出合适的算法模型，并在此基础上将算法整合进真实系统中，将理论运用到实践中，创造价值。

## 本文研究重点及工作安排

本文研究的重点主要有以下几个方面：

1. 文本分类算法的优劣以及实现。
2. 文本情感分析的实现。
3. 文本挖掘算法的工程化。

其中章节安排如下：

第三章将描述现阶段对文本分类以及情感分析的研究，同时确定需要比较和确定的文本分类算法。

第四章将主要介绍现阶段新闻资讯系统的架构，以及文本分类和情感分析在其中的位置。

第五章为实验部分，将对文本分类算法进行实验对比和调优，同时实现情感分析算法。

第六章为总结及工作不足和展望。

# 新闻资讯系统及文本挖掘研究现状

## 国内外新闻资讯系统研究现状

在97年以前，推荐系统主要表现为信息的过滤，例如垃圾邮件的过滤。当电子商务兴起后，涌现了更多的推荐系统。通过推荐系统预测用户的喜好，用户可以更方便的找到自己需要的产品。比较著名的有国外的Amazon、Ebay，国内的淘宝、京东等。

而近十五年来，传统媒体向网络的迁移以及新兴网络媒体的出现导致互联网上的新闻数量激增。人们搜寻信息的方式也变得越来越多样化，通过搜索引擎、RSS订阅等各种方式，来搜寻自己的东西。但是这个需求从没有得到完全的满足过。之后不断更新的搜索引擎算法，以及涌现出的各式各样的阅读平台、站点或软件。都是为了解决同一个问题——根据人们的需要来提供相应的信息。个性化的应用也相应的产生，而在这背后支持的是不断完善的推荐技术。十五年过去了，个性化推荐依旧是当今的研究热点，工业界与学术圈在此领域中都相当的活跃。

个性化推荐技术研究的核心问题，是预测一个用户对其未看到的事物的喜好程度，或者说是预测一个用户对一个新鲜事物感兴趣的概率。

通常来说，推荐算法一般分为三类：

1.基于内容的推荐——推荐的依据是事物本身的特征

2.协同过滤的推荐——推荐的依据是与用户相似的用户喜好的事物

3.混合推荐——将上述两种方法结合的推荐算法

资讯的个性化推荐也是基于上述的理论。本文列举了一些互联网中的资讯推荐产品，进行说明。

DIGG是一家致力于新闻挖掘的资讯平台。它结合了书签、博客、RSS 以及无等级的评论控制，它的独特在于它没有职业网站编辑，编辑全部取决于用户。用户可以随意提交文章，然后由阅读者来判断该文章是否有用，收藏文章的用户人数越多，说明该文章越有热点。

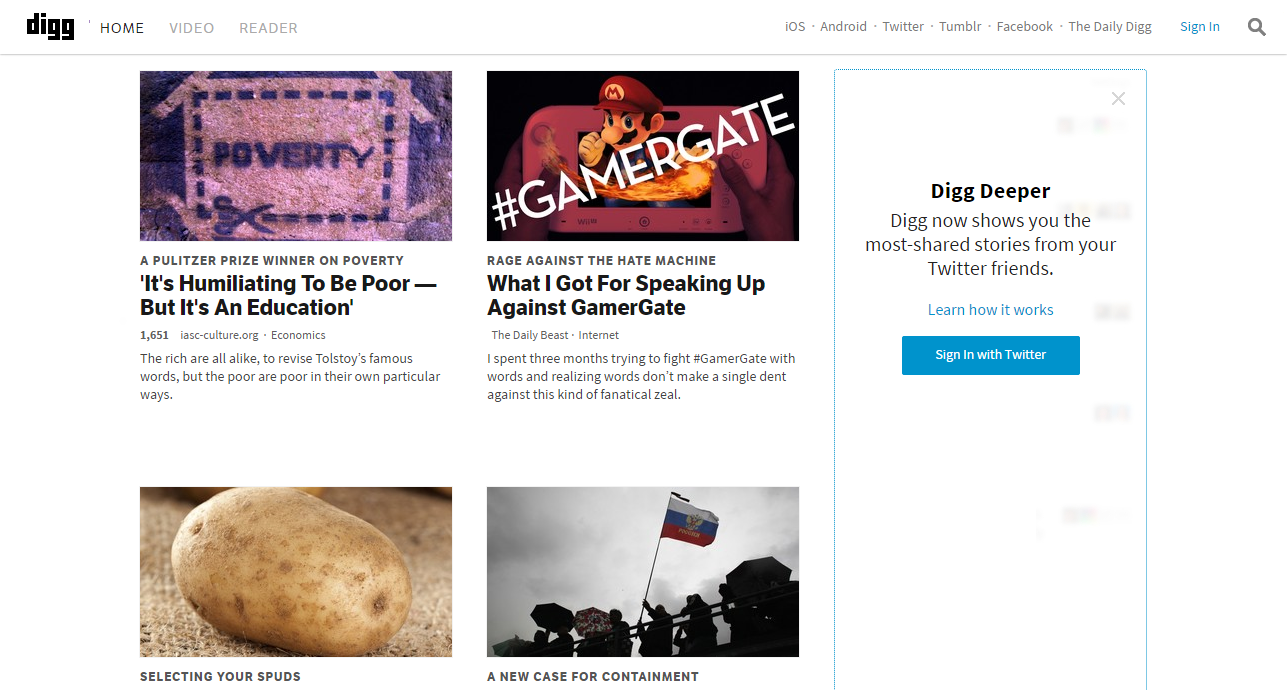


图 2.1DIGG官网

DIGG的推荐模式更多的是基于协同过滤的或者是混合模式的，其推荐的精彩内容多为其他用户所挖掘出来的热点或者是大家都比较感兴趣的话题。

极客公园是国内最大的创新者社区，其中汇聚了互联网产品和科技趋势的最新信息。其网络平台更多的是积累人气以及吸引流量的作用，而其更专注与线下活动，并且将每一次线下活动都进行包装、制作。



图 2.2极客公园官网

## 自动化文本分类研究现状

在机器学习兴起之前，文本分类工作主要有人工完成。例如路透社的基于知识工程的Construe 系统，其通过专家编写的分类规则，对每天上千件稿件进行自动化的分类。但是人工进行文本分类效率低下，并且错误率较高。在机器学习兴起之后，人们开始探索如何让机器代替人工进行繁重的分类任务，自动化文本分类诞生了。

目前英文的自动化文本分类已经取得了丰富的成果，并且提出了许多算用于自动化文本分类。比较经典的有贝叶斯分类器、SVM分类器、KNN分类器、神经网络分类器等。并且均有相当丰富的文献，常见于信息检索、机器学习等期刊杂志。97年，多特蒙德大学在文本分类中引入了向量空间模型，正式将机器学习引入自动化文本分类。98年，支持向量机（SVM）由Joachims应用在自动化文本分类中。之后，卡内基梅隆大学的Yiming Yang等人在文本分类中运用了决策树模型。随着研究的深入，人们开始研究更加细节的分类课题，例如特征维度的删选、基于深度学习的文本特征抽取等。

相对于英文文本分类的蓬勃发展，中文的文本分类技术起步则相对较晚。但是随着时间的推移，国内学者也取得了丰富的成果，相继提出了基于最大熵《文本分类若干问题研究》[55]、基于互依赖和等效半径的SECTILE分类算法《文本分类若干问题研究》[56]等等。

文本分类最初是机器学习和信息检索的交叉课题，而现在传统的文本分类已经开始走出实验室，被运用到各个实际的生产环境中，理论上可行的模型能否经得住实际应用的考验变得越来越重要。

## 情感分析研究现状

情感分析又称为意见挖掘[文本情感分析\_赵妍妍]，是文本挖掘中重要的一个部分。早在2001年，就有学者提出了情感分析之一概念，其主要运用在yahoo股票网站中股民的留言[《评论文本的情感分析研究及其在信用评估中的应用》[3]]。而在2002年，机器学习被运用到情感分析中，pang[《评论文本的情感分析研究及其在信用评估中的应用》[5]]等人发现，在电影评论中，SVM的效果比朴素贝叶斯、最大熵等算法好。

一般的，情感分析分为两类：基于词典的和上面提到的机器学习。基于词典的方法主要关注情感词在文本中的共现情况。通过计算情感的PMI来划定文本的情感倾向[[《评论文本的情感分析研究及其在信用评估中的应用》[6]]]。而基于机器学习的情感分析的成果则更加引人注目。将情感分析为题转化为分类问题，简化了情感分析的步骤，同时也能够将文本分类的成果直接应用上去，不失为一个非常好的做法。Ghose利用机器学习的方法训练出了一批判断正负倾向的分类器，并且运用在Amozon的商品评论中，以此来评判一款商品的火爆程度。

近几年来，文本情感分析在工业界中等到了一定程度上的应用。例如NEC公司提出的情感分析工具：ReviewSeer，它根据评论文本以此来判断商品的受欢迎程度。[《评论文本的情感分析研究及其在信用评估中的应用》[10]]。由Liu等人开发的Pulse系统则通过可视化的方式将文本的情感倾向展现出来，让人们更加直挂你的感受到其判断对象在客户中的受欢迎程度。

相对与国外的情感分析热潮。国内的情感分析则起步较晚。由于汉语表达多样化，以及中文句式相对英文句式更为复杂。这对中文情感分析造成了较大的困难和挑战。哈工大和复旦大学相对来说走在了中文情感分析的前列。叶强,李一军等人通过改进PMI，得到了和英文同类研究相似的成果，并提出了基于PMI-IR的中文语义倾向的情感分析方法[《评论文本的情感分析研究及其在信用评估中的应用》[15]]。复旦大学的朱嫣歲等人提出了基于语义相关场和语义相似度的方法。[《评论文本的情感分析研究及其在信用评估中的应用》[16]]

相对于自动化文本分类，情感分析现在依旧面临许多问题：

1. 如果文本中同时存在正反态度，并且进行了相互比较。
2. 数据过于稀疏，特征维度相对来说太少。不利于情感信息的抽取。
3. 跨领域问题。不同领域下，同一个词的情感程度可能是不同的，甚至极性都会随着领域的变化而改变。
4. 中文情感语料库相对来说比较匮乏。

# 文本挖掘技术基础

## 文本分类技术研究

### 文本分类任务

文本分类任务简单的描述为如下映射：

A为未标注类别的输入文本集合，B为输出类别集合又称A在对应类别中的映射，为分类器分类过程。

### 文本分类过程

文本分类器训练流程如下：

图 3.1 文本训练总流程

分类流程如下：

图 3.2 分类流程

接下来的章节将会着重描述文本分类的过程及相关步骤。

### 文本结构化

文本结构化又称文本向量化，是文本分类任务中最为重要的一个环节。

文本结构化的目的是让非结构化的文本数据在尽可能的减少信息损失的情况下转化为计算机可以识别并计算的数据格式。一下将对文本结构化过程中的几个重要步骤进行展开。

#### 文本表示

文本结构化最重要的目标就是将文本表示为计算机可以理解并且运算的数据形式。

现在主要有三种文本表示模型：

1. 概率模型
2. 代数论模型
3. 集合论模型

其中代数论中的向量空间模型是应用最广泛的文本表示模型。由XXXX#需要提供参考文献#提出，并且广泛应用于信息检索领域，并且取得了不错的效果。

本文将主要基于向量空间模型对文本进行结构化。

#### 最小语言特征单元提取

最小语言特征单元（以下简称特征单元）提取主要要目的是从文本中抽取出尽可能小，包含尽可能多信息的粒度单元。在中文文本处理中，这个过程又称为分词。

在文本挖掘的兴起的英语世界，特征单元字面上就是英文单词，与之所对应的是中文的词，而中文不像英语每个单词都是独立不可再分的，换而言之相对与英文单词，中文的单词没有明确的边界。

中文中特征单元粒度最小的是单字，虽然其足够小，能够最大程度上减小稀疏数据的产生，但是其所包含的语言信息太少，并不符合上述分词的要求。但是近期也有国外学者对中文文本分类做了部分研究，提出了根据单字构造空间向量模型，并且将其应用至亚马逊买家评论以及文本分类，并提高了准确度[1]《Text Understanding from Scratch》arXiv:1502.01710v1 [cs.LG] 5 Feb 2015。

值得一提的是，李景阳[2]在《文本分类中的特征选择与权重计算》中提到，特征单元维数在50000以上时，二字串的分词性能更加优良。但是考虑本文的应用场景是针对新闻资讯系统，每天更新的文章数量有限，相对于样本数量，维数过大会造成统计学中的维度灾难，同时会造成模型的过拟合，反而会降低分类器的性能。

现有的中文分词工具有如下：

表 3.1 分词工具比较

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 分词原理 | 分词速度 | 分词准确率 | 特点 |
| Anjs\_seg | HMM（隐马尔科夫模型）、CRF（条件随机场） | 200万字/秒（大约4000K/S） | 96% | 开源分词工具。有三种模块，base模块速度非常快，能达到300万字/秒，但不支持自定义词典；to模块速度和效果较为平衡，同时支持自定义词典和人名识别；nlp模块效率降低，但在分词效果上进一步增强，同时支持机构名识别和新词发现 |
| ICTCLAS | HHMM（层叠隐马尔科夫模型） | 25万字/秒（大约500K/） | 98.45% | 有开源版本和商业版本，分词效果非常好，但是相对性能较低，支持自定义词典、人名识别、机构名识别、新词发现等，以及语义理解等高级应用。 |
| Jieba | 基于Trie树结构实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图（DAG）；采用了动态规划查找最大概率路径, 找出基于词频的最大切分组合；对于未登录词，采用了基于汉字成词能力的HMM模型，使用了Viterbi算法 | 1.5MB/Second in Full Mode  400KB/Second in Default Mode | 98.45% | 在Python编程领域的高准确率、高效率的分词组件 |
| CRF中文分词器 | CRF（条件随机场，目前分词效果最好的一种算法） | 350万字/秒 | 95% | 开源包中的分词知识库较小，仅用了200万字的分词精度较低，仅供学习Crf分词算法使用。 |

本文对分词的原理不做过多讨论，使用的分词工具是基于jieba分词的jiebaR。

分词得到的语言最小单元在一般情况下，数量为会非常的大。其中包含了大量的噪音，在之后的工作之前需要对分词结果进行一轮粗略的清理，目的是清除例如“的”这类广泛存在于文本中，却又没有实际含义的词。这类词被称作为停用词。现如今已经有许多前人总结出了中文中的一些停用次，本文中用到的停用词整合了哈工大停用词表、四川大学机器智能实验室停用词库、百度停用词表。

#### 特征选择

特征选择是对分词结果进行降维的过程，在分词并且去除停用词之后进行。一般来说，每一个分类器都需要进行特征选择，目的是进一步减少特征单元的数量，降低计算压力，提升分类器性能。

一般来说有如下特征选择的方法：

1. 文档频度（df）

文档频度是最为传统的特征选择方法，但是大量实验表明，文档频度有着出人意料的效果，对于提升分类器性能，降低计算压力效果较好。本文将采取df > 2的方式对语料统一截断并降维。

1. 卡方检验（chisq test）

卡方检验源自统计学，其意义是度量某一变量与另一变量的相关程度，是显著性检验的一种。

其一般形式计算对象为2·2的混淆矩阵：

表 3.2 卡方检验混淆矩阵

|  |  |  |  |
| --- | --- | --- | --- |
|  | 属于类别c | 不属于类别c | 总计 |
| *包含特征词t* |  |  |  |
| *不包含特征词t* |  |  |  |
| *总计* |  |  |  |

卡方统计量计算公式为：

其中表示，既属于类别c又包含特征词t的文档频数；

表示，既不属于类别c又不包含特征词t的文档频数；

表示，包含特征词t的文档频数；

表示，不属于类别c的文档频数；

为所有文档数量。

其他以此类推。

由于其同时考虑的正例和负例，具有较强的现实意义。并且其是特征选择性能最为优良的指标之一。《文本分类中的特征选择与权重计算》[1,48,52]

1. 互信息

互信息源于信息论，描述了一个随机变量中包含的关于另一个随机变量的信息量这样一个指标。其数学公式为：

互信息在文本特征为度筛选是也有广泛的应用，在本文中，不做考虑。

1. 信息增益（information gain）

信息增益同样来自信息论。其最典型的特征是非对称的，即P对Q的增益量Q不等于Q对P的增益量。其数学公式如下：

广泛的实验证明，IG同样是性能良好的特征维度删选指标之一，但较为偏向高频特征。《文本分类中的特征选择与权重计算》[1,52]

#### 权重计算

常用的权重计算方式有三种:

1. 布尔型

布尔型是最基础的权重表示方式。如果类别ci存在特征ti，则wcti为1，否则，wcti为0。由于布尔权重会丢失大量的文本信息，所以在本文那种不对其作过多的描述和进一步的实验。

1. 频度、词频（term frequency）

词频是最常用的权重计算方式之一，源自统计学，定义为一个词在一篇文档中出现的频数。其认为，一个词在一篇文档中出现的频数越高，则其重要程度越重。由于计算方便，以及优良的分类性能，成为了工程实践中分类模型权重的不二之选。

1. TF-IDF（term frequency–inverse document frequency）

TF-IDF源自于信息检索，描述词频高的词对于一个语料库的其中一个文件的重要程度。数学公式为：

# 数学公式 #

很明显，TF-IDF由两部分组成，词频tf和逆文档频度idf相乘。tf在b.已经讨论过，而idf主要描述一个词在逆向文档中的频度，主要考量一个词在一个特定的语料库中的重要程度。一般来说，词的TF越大则说明该词越重要，而IDF越大说明在语料库中该词分布月普遍，相对的重要性越低。在本文中，TF-IDF指标由于其特殊性——需要对整个语料进行计算，导致了工程实施上的障碍。对于新获取的待分类的文章，其中词汇的TF-IDF，应该与训练集中的词一同计算才能保证其准缺性，这样就导致了一个问题——每一次分类之前都需要重新计算一边权重，接着影响到模型的训练，这样会严重制约分类效率，并且不利于模型维护以及更新。所以在本文中，TF-IDF权重只会作为一个重要的实验参考对象，来评判其他权重的分类性能。

### 文本分类算法

文本分类的算法模型大多基于传统的分类模型。在本问提到的新闻资讯系统中，类别是认为指定并且确定的。所以采用的是有监督的机器学习算法。

从概率模型的角度来说，常用的有监督的机器学习算法主要有两种：

基于概率的产生式模型（generative model）[《文本分类中的特征选择与权重计算》]，其主要代表有朴素贝叶斯模型，高斯模型，贝叶斯网络等，限于本文的应用场景，除了朴素贝叶斯模型外，其他模型不做过多的阐述和介绍。

朴素贝叶斯模型已著名的贝叶斯定理为基础，在文本分类中可以描述为

遵循经验风险最小原则，其鲜明的特点是运用了：先验知识（先验概率）以及模型可以增量式学习[《文本分类中的特征选择与权重计算》]，但是由于贝叶斯定理的一个前提假设，变量之间必须满足独立性原则，使其在文本挖掘中缺乏足够的理论支持。因为文本中的词不是相互独立的，其间包含了语义关系。同时，基于贝叶斯理论的模型需求大量的训练集，以寻求更加逼近真实的概率分布。在少量样本的的情况下，其分类性能会大幅衰减。但是在实际使用中，朴素贝叶斯的效果是值得称道的，在本文中，朴素贝叶斯是一个待测试的分类算法。

判别模型，也称非参数模型。其理论基础源自于统计学习理论，提出了结构风险最小的思想。典型代表是SVM，逻辑回归等。

SVM模型的思想为将现有维度下不可分的问题映射到高维再进行分类，从而解决当前维度下类别不可分的问题。其中距离分类界限最近的向量，被称为支持向量。#SVM分类图例一张#

在一般的传统统计模型中，关注的点在于经验风险最小，这使得传统模型面临一个问题，就是过拟合和不可扩展。为了解决这个问题，SVM模型于上个世纪被踢出，其考量的是结构风险的最小化。这个特性使得其具有较好的泛化能力。

SVM的理论较为深刻，而在本文的背景下，更多的是应用场景，对其理论描述不做过多的讨论。其中的参数着重于核函数、gamma和损失函数cost。在试验中，将其设置为如下范围进行考量。

其中核函数备选两个：线性核函数和高斯核函数。

本文中拟定的gamma取值范围10^-6到10^-1，cost取值范围为10^-2到10^2

gamma参数隐含的决定了kernel为radial时，数据映射到高纬特征空间后的分布，而cost又称C或损失函数，度量了对误差的容忍度，越大表明越不能容忍出现误差。

本文将着重对SVM进行阐述和分析，并且在实验中作为被试，对比如其他分类算法。

值得一提的是，本文背景中还有一个已经使用了的分类模型，基于关键词的匹配分类模型。其理论基础非常简单。即每一个类别中都有相应的关键词，在分类过程中，对文章的标题进行关键词的匹配，标题中出现了类别关键词的即被判为这一类别。这个模型可以看作是路透社专家分类的简化版，其分类规则约减为关键词。而制约这个分类算法性能的是关键词的质量、匹配的对象。

在本文应用场景中，关键词来源于idf算法的关键词抽取，即对每一个类别的文本中的词均计算idf，统计整合其中词频较高的，同时人工干预，剔除一些过于宽泛的词，保留类别辨识度较高的词；类别间的距离较小，隶属于IT行业下的二级或者三级分类，甚至有的仅仅是话题，而关键词匹配却有着出乎意料的分类效果；在本人经过大量试验后发现，对于关键词匹配的分类，基于文章标题的匹配效果是最好的，原因在于新闻中，文章标题大多都已经包含了该新闻所属的类别信息。而文章内容对于匹配来说有着太多的不确定性，会使类别匹配过于杂乱。

本文对这个算法的效果和SVM以及朴素贝叶斯进行比较。

### 文本分类性能指标

文本分类指标参考的是信息论中的命中率。

和卡方检验的矩阵一样，本文性能指标计算也依托于混淆矩阵，其在信息论中的形式如下：

表 3.3 性能检测混淆矩阵

|  |  |  |
| --- | --- | --- |
|  | 实际命中 | 实景未命中 |
| *检测到* | TP | FP |
| *未检测到* | TN | FN |

其中

TP，表示正确命中。

FP，表示错误命中。

TN，表示正确未命中。

FN，表示错误未命中。

本文中需要关注的两个指标是准确率（P）和召回率（R）。其计算公式如下：

同时还有F-measure，这个指标同时考虑了准确率和召回率，计算公式如下：

其中为加权调和系数，其大小控制了F检验量偏向于召回率R还是准确率P，当为1的时候，表示F统计量均衡考虑召回率和准确率。

另外，本文将引入准确率和召回率的ROC曲线，尽行可视化的分类器性能比较。

## 情分析技术研究

### 情感分析种类

1. 基于词典的情感分析

这是比较传统的情感分析方法，其关键是对于情感词的人工删选和标注。这也是本文所关注的情感分析方向。

1. 基于机器学习的分类

是现在比较常用的情感分析方法，其把情感分析转化为分类问题，进而用机器学习的方法来解决问题。其优点和上述文本分类中的有点相同，但同样的，其缺点在于泛化能力受到训练集的制约。在本文文本分类的基础上，将不对这种方法做过多的讨论。

### 情感分析流程

与大部分文本挖掘流程相似，在正式分析之前，都需要进过文本的预处理，流程如下：

图 3.3情感分析总流程

对于非机器学习的情感分析部分还能继续细化为：

图 3.4 情感分析细节流程

其中情感信息的抽取是最为底层也是相对重要的工作，一般的大多基于情感词典来完成。

### 情感分析算法

情感信息权重计算是表征文档情感的重要部分，基于PMI的SO正负极性情感算法公式为：

首先计算一篇文档的关于情感的PMI，然后对所有情感的PMI进行算数加和，得到文档的最终情感倾向。

在本文中，情感分析算法的实现主要基于贝叶斯理论。

首先计算一篇文章中每一个情感类别所包含的情感词在情感词典中的先验得分，公式如下:

接着计算每一篇文章中，每一个情感的后验得分

为最终的情感得分，通过比较每一个情感类别的情感得分，取最高的为最终的情感结果。

而极性得分的计算和情感得分计算步骤一致。

## 本章小结

文本分类和情感分析的种类繁多，各种算法的性能也不尽相同。本文在接下来的第五张将会着重对文本分类进行实验，并且根据性能指标对每一个分类算法进行评判和比较，挑选出最优的酸饭用于实际的系统中。

# 新闻资讯系统架构及功能

## 新闻资讯系统架构设计

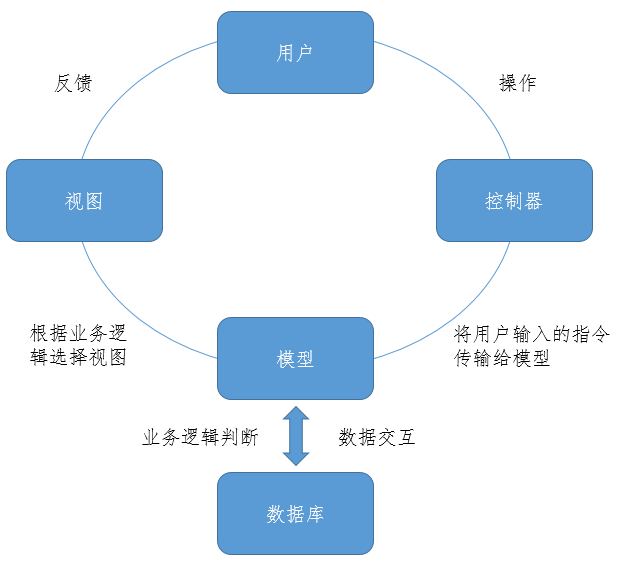


图 4.1 系统MVC架构图

本文算法需要集成的系统架构主要以MVC为主，而本文涉及到的文本挖掘算法算法多为离线静态计算，没有在线实时计算的压力，具体流程如下：



图 4.2 个性化新闻情报系统流程架构

## 新闻资讯系统模块设计

### 爬虫系统设计

爬虫系统是整个资讯系统的基础，其工作主要目的是从互联网中根据网站列表，抓爬并解析网页内容。其中爬虫系统的结构如下：

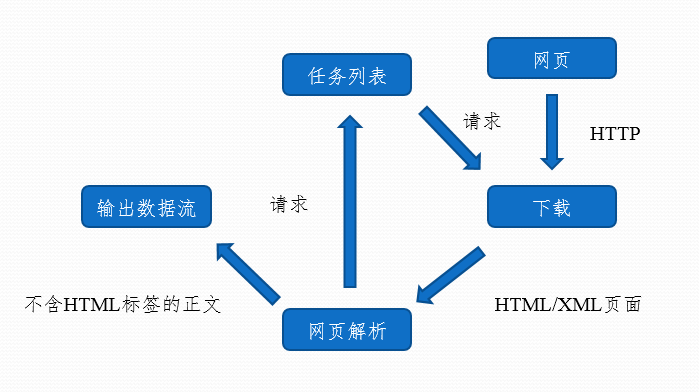


图 4.3 爬虫系统架构图

在这个架构中，最为重要的部分是爬虫的任务列表流程，其执行机制和执行效率直接影响爬虫的运行效率和效果。其流程如下图所示：

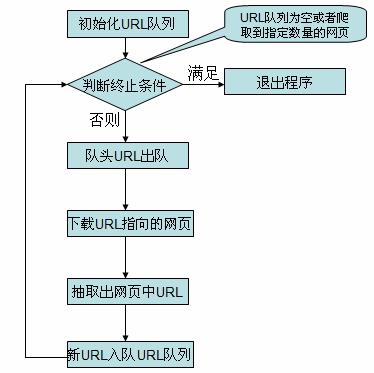


图 4.4 爬虫任务列表流程

### 数据库设计

数据库是文本系统中的核心部分，分为爬虫数据库、页面展示数据库和中间数据数据库。

爬虫数据库：其职能主要是对经过爬虫系统处理的网页内容进行存放，并于ETL程序相连接，定时抽取并清理数据至页面展示数据库中。

页面展示数据库：其职能主要存放各种服务以及应用算法的结果数据，供前台页面直接调用。

中间数据数据库：主要存放一些中间数据，例如用户行为记录等前台不显示但是对于推荐有影响的数据，供个性化推荐算法使用。

其中页面展示数据库和中间数据数据库的数据库维度表关系如图4.2所示：

user、user\_preference、user\_configure、email\_reset\_psd、user\_reading、user\_appraise、user\_mark为用户行为记录相关的表；category、subscription、tag、article、article\_classified、article\_result、article\_categories为文本相关表。

其中article表为最为重要的数据存储表，主要存储清理过的文本信息。所以针对这张表做了特定优化。对ID字段建立索引，并对article表进行分区建表。

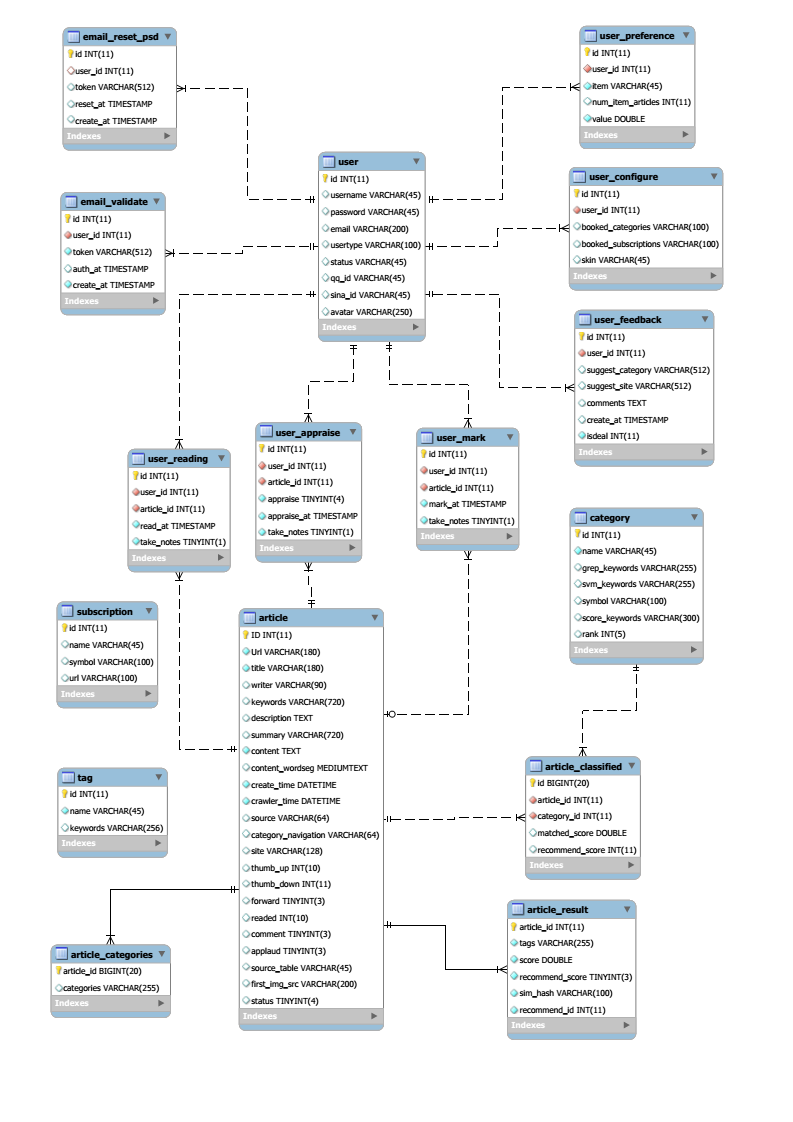


图 4.5 数据库设计

文本存储表article表结构如下表所示：

表 4.1 Article表结构及字段

|  |  |
| --- | --- |
| **Column** |  |
| ID | int（11） UN AI PK |
| Url | varchar（180） |
| title | varchar（180） |
| writer | varchar（90） |
| keywords | varchar（720） |
| description | text |
| summary | varchar（720） |
| content | text |
| content\_wordseg | mediumtext |
| create\_time | datetime |
| crawler\_time | datetime |
| source | varchar（64） |
| category\_navigation | varchar（64） |
| site | varchar（128） |
| thumb\_up | int（10） UN |
| thumb\_down | int（11） |
| forward | tinyint（3） UN |
| readed | int（10） UN |
| comment | tinyint（3） UN |
| applaud | tinyint（3） UN |
| source\_table | varchar（45） |
| first\_img\_src | varchar（200） |
| status | tinyint（4） |

### API设计

API主要负责连接应用层和数据层，并提供诸如get、post、delete等方法，以实现前后台数据交互、用户行为记录等各种功能。

本文系统的API主要实现形式如下：

表 4.2 API设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 根目录 | http://IP:port/htnewsroom | | | |
| 使用说明 | 将资源地址{XXX}替换成对应的id或者名字，e.g. /articles/article-{aid} ==> /articles/article-3 表示请求id为3的文章内容 | | | |
| 类别 | 请求方法 | 资源地址 | 参数 | 简介 |
| article | GET | /articles/article-{aid} |  | 获取单篇文章--依据文章id |
| /articles/user-{uid}/recommendations | page=$ & size=$ | 获取精选文章--根据用户id |
| /articles/user-{uid}/categories | page=$ & size=$ | 获取主题推荐文章--根据用户id |
| /articles/user-{uid}/categories/category-{cid} | page=$ & size=$ | 获取单个主题文章--用户id和主题id |
| /articles/websites | page=$ & size=$ | 获取第三方源文章 |
| /articles/websites/website-{sid} | page=$ & size=$ | 获取单个第三方源文章--依据源id |
| /articles/tags/tag-{tname} | page=$ & size=$ | 获取单个标签的文章--依据标签名字 |
| /articles/hot | size=$ | 获取热门文章 |
| /articles/user-{uid}/mark | page=$ & size=$ | 获取用户收藏文章-- 依据用户id |
| POST | /articles/user-{uid}/article-{aid}/like |  | 用户点赞 |
| /articles/user-{uid}/article-{aid}/mark |  | 用户收藏 |
| /articles/user-{uid}/article-{aid}/hate |  | 用户点踩 |
| /articles/user-{uid}/article-{aid}/read |  | 用户点击阅读 |
| /articles/user-{uid}/article-{aid}/share |  | 用户分享 |
| DELETE | /articles/user-{uid}/article-{aid}/likeCancel |  | 取消用户点赞 |
| /articles/user-{uid}/article-{aid}/markCancel |  | 删除用户收藏的某篇文章 |
| category | GET | /categories |  | 获取主题对象列表 |
| /categories/user-{uid} |  | 获取用户订阅主题列表 |
| website | GET | /websites |  | 获取第三方文章来源列表 |
| /websites/user-{uid} |  | 获取用户特定的第三方来源列表 |
| tag | GET | /tags |  | 获取所有标签 |
| /tags/user-{uid}/hot |  | 获取热门标签 |
| user | GET | /users/user | username=$ | 获取当前用户 |
| /users/user-{uid}/configure |  | 获取当前用户的配置信息 |
| /users/user-{uid}/article-{aid}/status |  | 获取当前用户对文章的状态--点赞、收藏 |
| POST | /users/user-{uid}/feedback | suggestCategory & suggestSite & comments | 添加当前用户的反馈 |
| /users/user-{uid}/configure | categories="id1,id2,id3" & websites="id1,id2,id3" | 添加或修改当前用户的订阅主题和订阅源 |
| search | POST | http://IP:port/jdbc/jdbc/\_search | from=$ & size=$  再加上json格式的  字符串 例如  {"query":  {"query\_string":  {"query":"title:  云计算和大数据的发展规"}}} | 文章相关推荐API（默认按相关排序） |
| http://IP:port /jdbc/jdbc/\_search | 搜索API（按文章相关性排序） |
| [http://IP:port /jdbc/jdbc/\_search](http://IP:port%20/jdbc/jdbc/_search) | 搜索API（按文章发布时间降序） |
| [http://IP:port /jdbc/jdbc/\_search](http://IP:port%20/jdbc/jdbc/_search) | 搜索API（按文章发布时间升序） |
| [http://IP:port /\_analyze](http://IP:port%20/_analyze) | analyzer=word&text=$ | 搜索内容分词API |

### ETL及文本分类

#### ETL的功能和实现

ETL是Extract translate and loading的缩写，其作用是抽取各方面的源数据，经过特殊处理转换后，以约定形式保存起来的一个过程。

在爬虫阶段，数据源是种类各异的网站站点，其显示正文内容的风格也千差万别；同时也可能出现抓爬到资讯以外的网页。因此，需要ETL过滤不符合规范的文章，统一正文的数据结构，以方便后续处理（如正文的显示）。

ETL的功能如下：

1. 过滤掉title、url为空的数据。
2. 过滤去掉所有html标签的正文内容长度少于150个字符的文章。
3. 过滤掉正文中的特殊字符，如全角空格、回车、换行等。
4. 清除<p>标签前后的空格，控制页面显示的缩进。 例如：<p> aa </p>转化为<p>aa</p>。
5. 过滤类似<p></p>的空标签。
6. 使用jsoup解析正文content，为正文内容添加合适的标签。
7. 为没有p标签的段落加上p标签。
8. 为img标签前的p加上class=’article-img’。
9. 为原始的图片url 生成映射，让外网可以获取到图片。

ETL流程如下图所示：

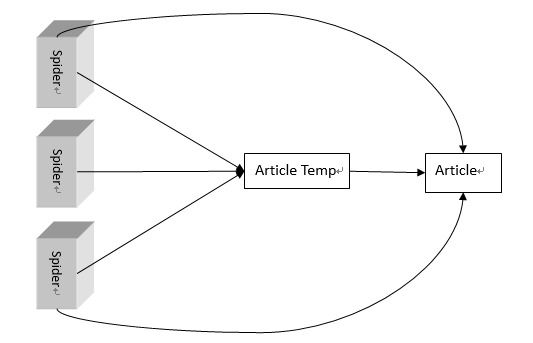


图 4.6 ETL流程图

ETL每隔两小时，从爬虫数据库中进行数据抽取和清理。并且在ETL过程中，对文本进行初级的查重控制，其主要机制如下：首先对源数据的每一篇文章，都在article中查询是否已经存在，存在就忽略。否则，将源数据按照ETL的规则统一存进临时数据库表article\_temp，然后在一次ETL结束后将article\_temp中的所有数据存进目标数据表article，然后将article\_temp的数据清空。

实际效果证明，这样的ETL设计是合理且运行效率是可以接受的。

#### 文本分类的功能和实现

文本分类的功能主要是对ETL之后的文章进行一对话题分类，并对每一篇文章打上类别表情，存入数据库中，供前台调用，为用户呈现出不同的话题。

文本分类紧接着ETL之后。其实现流程如下：

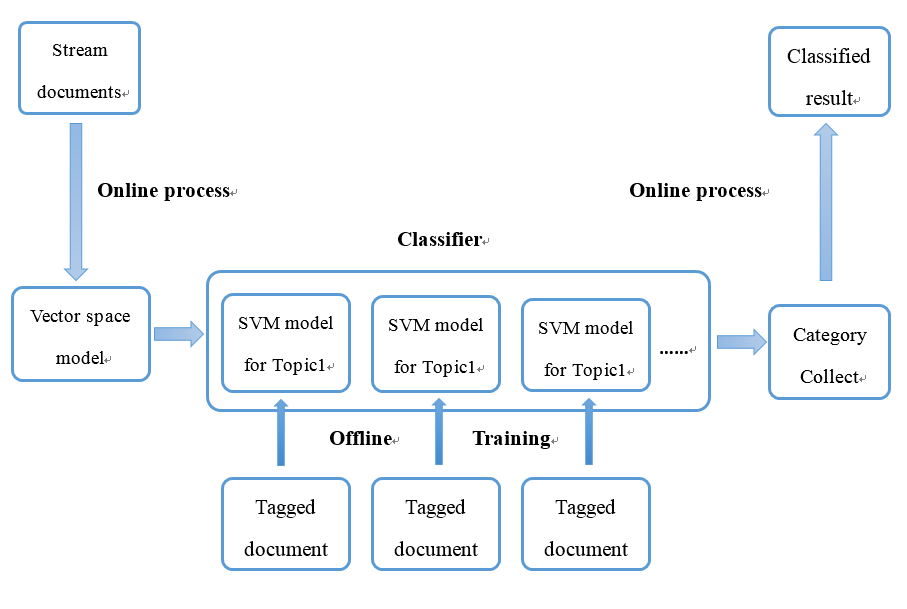


图 4.7 文本分类流程图

本文系统中，文本分类如上图所示，为离线处理过程。时间设定在每次ETL结束之后。并且本文系统中，文章和类别的关系是一对多，所以分类信息和文本信息分表存储。

分类信息表字段如下：

表 4.3 Article\_classified表结构

|  |  |
| --- | --- |
| Columns**:** |  |
| id | bigint（20） AI PK |
| article\_id | int（11） |
| category\_id | int（11） |
| matched\_score | double |

其中article\_id对应article表中的文章id，category\_id对应category表中的类别id，match\_score为这篇文章属于这个类别的程度得分（暂未实现）。

### 热点抽取及情感分析

热点抽取和情感分析的功能为：为舆情监测系统提供热点话题和媒体对热点话题的情感倾向。

情感分析流程：

本系统中，热点抽取是离线作业的，时间周期分为一天一循环和一周一循环。

每天循环的热点主要计算前一天的热点新闻，并且记录热点新闻的时间节点；每周一循环的热点抽取主要计算上一周的热点话题。

情感分析在热点抽取之后进行，对热点抽取后所的到的热点话题文章进行倾向性分析和情绪分析，并汇总和输出文本的情感倾向已经极性。

### 搜索

搜索主要提供对article表的全文搜索支持。

搜索功能架构：

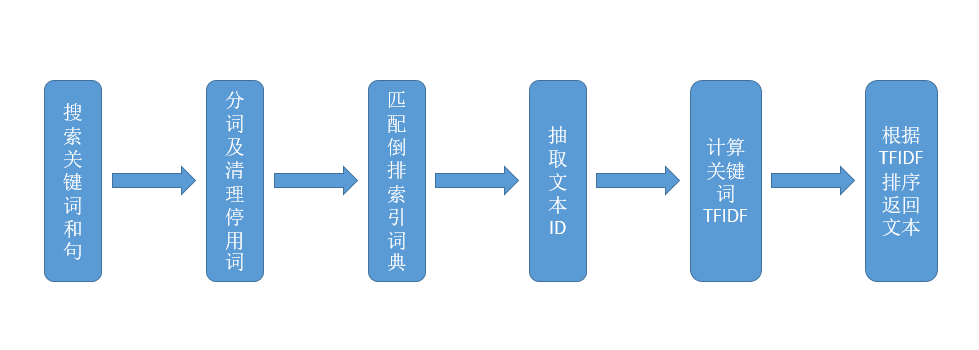


图 4.8 搜索框架流程

其中搜索功能的关键技术在于建立文本的倒排索引。本文系统中倒排索引是离线计算的最后一步完成的，即情感分析结束之后，对article表进行倒排索引。流程如下：

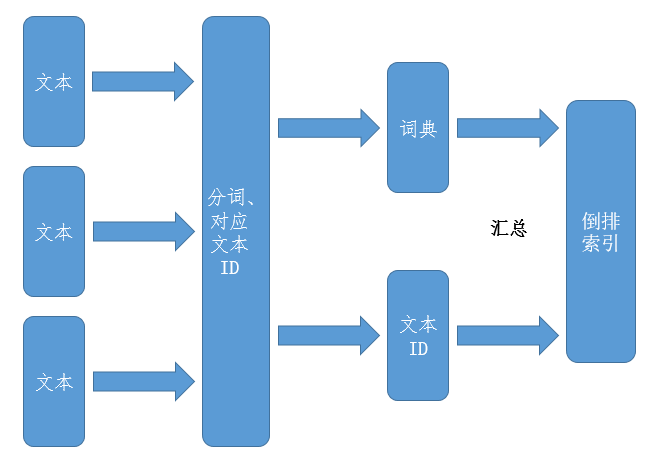


图 4.9 倒排索引建立流程图

为保证搜索引擎工作的稳定性，搜索倒排索引单独建立在非数据库服务器上，保证日常应用的抗压能力。

### 前端页面

前端页面直接面对终端用户，其主要展示功能如下：

1. 展示主题（类别）、以及订阅源

首页展示默认几个的新闻资讯主题，每个主题下可以看到推荐的文章，包括每篇文章的标题、发布时间和来源。

用户可以点击每篇感兴趣的文章，进入到原网站查看具体资讯内容。

首页的每个主题下可以看到所有主题的文章，通过瀑布流的方式展现，效果图如下，其中红框标识的为主题也就是类别站选项，与之对应会在右边主窗口以瀑布流形式展示相应的文本：



图 4.10 个性化新闻资讯系统首页

1. 热门话题推荐页面

对于普通匿名用户在首页上还可以看到热门话题推荐以及热门新闻资讯推荐，该板块并不针对某一行业，而是为一般用户提供一些目前网上流行的热门话题和热点新闻，或者用户所有较多的、关注较多的话题。

示例如下：

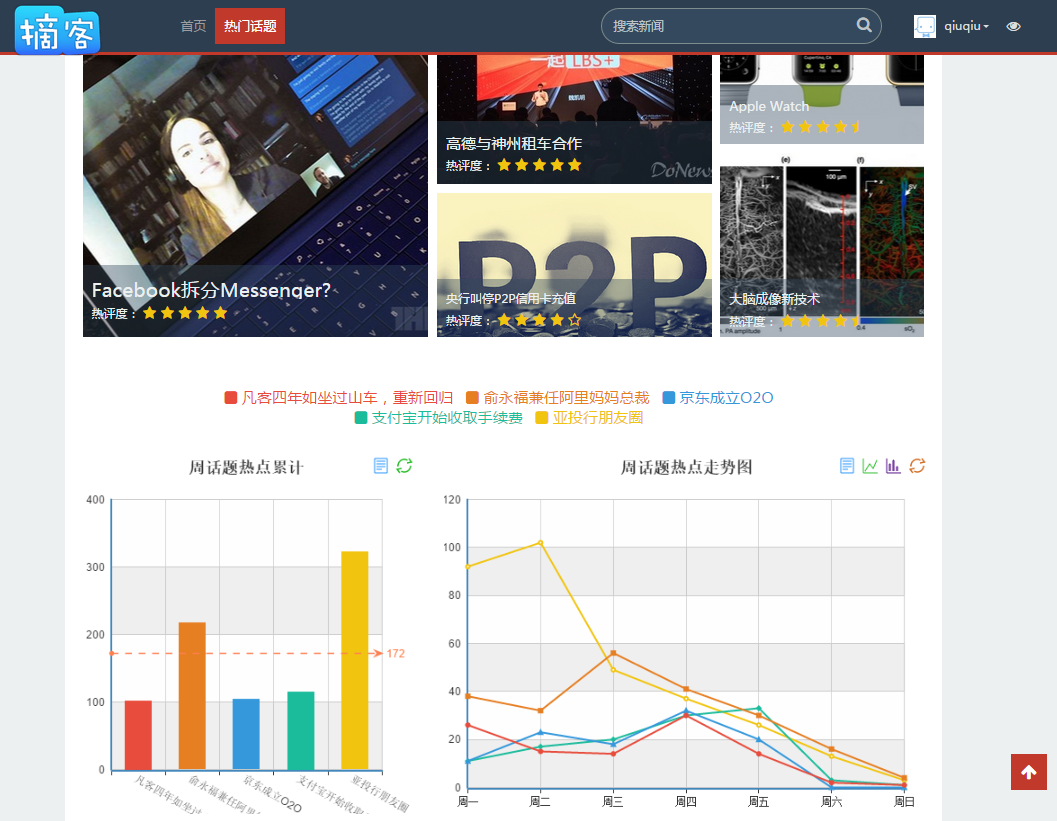


图 4.11 热门话题展示

1. 文章搜索页面：

在主页面上有一个搜索框，匿名用户可以搜索任何关键词的文章。输入搜索关键词后，回车直接跳转到搜索页面。搜索页面将显示搜索出来的文章记录总数，搜索出来的文章标题、描述、文章来源、文章发布时间。

效果页面如下：



图 4.12 首页搜索框

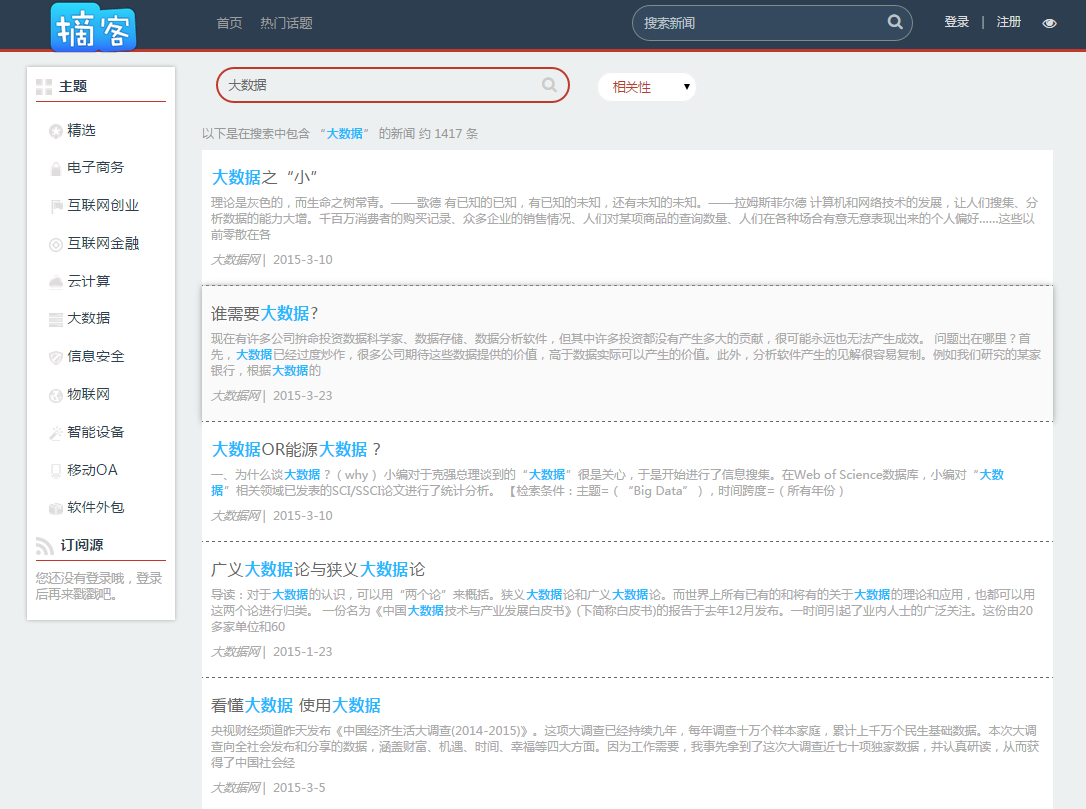


图 4.13 搜索结果页面

1. 登陆注册功能

登陆页面如下图所示，主要提供2种登录方式：注册账号登陆、社会化第三方登陆。

1. 对于登陆用户提供个性化的信息配置页面

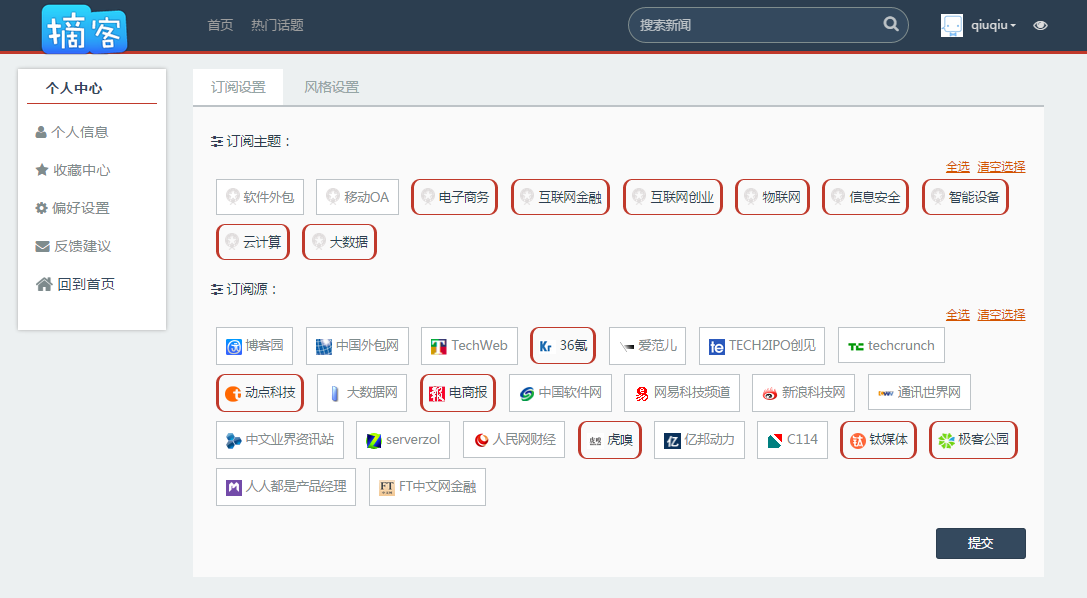


图 4.14 个性化配置页面

### 个性化推荐

个性化推荐是本系统中不多的需要实时在线计算的服务。其基于两部分组成，用户的行为记录和文本的阅读信息。

通过前台对用户行为记录，对用户行为进行建模，抽取用户站点偏好、用户文本类别偏好、用户文本TAG偏好，同时考虑文本的时间、文本阅读数、文本点赞数。将各部分结果加权计算，并对最终的得分进行排序，优先推送得分较高的文章给用户。

表 4.4 个性化推荐权重明细

|  |  |  |
| --- | --- | --- |
| 权重意义 | 权重系数 | 权重得分取值范围 |
| TIME\_WEIGHT | 0.3; | （-inf,10] |
| READ\_WEIGHT | 0.2; | [0,10] |
| APPRAISE\_WEIGHT | 0.2; | [0,10] |
| SITE\_WEIGHT | 0.1; | [0,10] |
| CATEGORY\_WEIGHT | 0.1; | [0,10] |
| TAG\_WEIGHT | 0.1; | [0,10] |

本文中的个性化推荐算法是简单的线性加权，并没有用到一些常用的商品推荐网站所用到的关联分析或者协同过滤。

使用这样一个简单算法的原因有以下几点：

1. 满足实时计算的性能要求
2. 模型的建设较为简便和系统维护成本较低，适合前期平台搭建。
3. 文本推荐相对于商品推荐的效用成本更低，用户在阅读文章时，其兴趣捕捉波动更大，并且用户本身也并不介意一些他并不是很感兴趣的文章出现在他的视野中。

总之，以上个性化推荐的算法是适合本文个性化资讯系统当下的要求的。

## 本章小结

文本分类处在服务层，情感分析处在应用层。其中，分类算法只与数据库相连，几个离线计算流程如下：

两小时为周期的离线计算流程，

图 4.15 2小时离线流程

每天离线计算流程，

图 4.16 每日离线计算流程

每周离线计算流程，

图 4.17 每周离线计算流程

可以看到，每一个离线计算流程中中，或者包含文本分类，或者包含情感分析。文本分类和情感分析对于整个系统来说是比较重要的业务实现模块。

# 文本分类及情感分析实验

## 文本分类比较实验

### 实验环境设计

本文所涉及的实验环境均为本机系统上，系统配置如下：

CPU：I7 4790K@4.5GHzX4

主板: INTEL Z97 芯片组

硬盘: 创见240G SSD

内存: 8G@2400MHZ

本文使用的分析工具是R语言。R语言是一款出色的统计软件，其不断更新的CRAN库彰显了这款软件的生命力。现如今，CRAN社区上发布的包超过了7000个，其中包含了大量非常优质的统计学和数据挖掘算法。就包括了本文中所用到文本挖掘包tm、SVM和贝叶斯分类包e1071、非结构化数据处理包slam和稀疏矩阵包sparseM。其中e1071种的SVM模型是封装了台湾大学教授林智仁等开发的libsvm工具，非常的方便。

但同时R也有其弱点，作为一个非工程实施语言，在工程中与其他语言的粘合度不高，甚至需要依附于其他语言来调用，例如PYTHON或者JAVA。所以本文在上章节中提出了MVC的系统设计典范，将模型部分R代码独立出来，通过linux shell脚本定时调用来和其他功能模块结合。

### 文本分类数据来源

本文的文本分类数据均来自互联网，通过爬虫系统没天定时增量抓爬、经过ETL的清理得到。

总共6000篇文本。划分训练集4500篇，测试集1500篇。通过人工标注拟定的类别有云计算、大数据、性能测试、电子商务、互联网金融、互联网创业、信息安全、物联网八个类别。在分词结束后，数据集的原始特征词维度达到了70000维。为了防止过拟合，需要进行降维处理，为了找到合适的维度数目，会对设置不同维度选择的阈限（卡方检验量的值）。

### 实验结果分析

首先对语料进行文档频度df ≥ 2的截断，保证每一个词特征至少出现在任意两篇及两篇以上的文章内；其次计算每一个词的tfidf，并设置tfidf阈值为0.05，去除噪音词。在这些工作之后，将预料库进行卡方检验，得到如下表：

表 5.1 类别特征维数

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ≥1 | ≥1.5 | ≥2 | ≥2.5 | ≥3 | ≥3.5 |
| 云计算 | 1253 | 958 | 783 | 713 | 575 | 538 |
| 信息安全 | 1496 | 1240 | 846 | 741 | 698 | 293 |
| 电子商务 | 1694 | 1240 | 1049 | 561 | 450 | 394 |
| 互联网金融 | 917 | 793 | 669 | 613 | 590 | 523 |
| 互联网创业 | 724 | 595 | 525 | 476 | 433 | 399 |
| 大数据 | 1034 | 867 | 748 | 665 | 564 | 534 |
| 物联网 | 830 | 705 | 626 | 567 | 501 | 434 |
| 性能测试 | 2055 | 1669 | 1432 | 1289 | 1189 | 1148 |

可以观察到，随着卡方值阈限的提高，每一个类别的特征为度均在减少，当然这非常好理解，但是本文更加关注一点，即维度下降的速度。考虑这一点的原因在于，如果特征维数下降速度越来越快说明有价值的信息主要堆积在卡方值较高的水品上，对于这种情况，需要将卡方阈限适当提高，确保无效信息较少的被纳入特征维度；相反，如果特征维数下降越来越慢，说明无效信息都堆积在前方，这时候需要适当降低阈值，以保证尽可能多的包含有效特征。

图 5.1 维度递减

图 5.2 维度变化趋势

可以发现，特征维度数量的减少不是线性的，随着卡方阈限的线性提高，大多数的类别递减速度是越来愈缓慢的，除了电子商务是波动的、信息安全在1.5之后才开始衰减。

值得关注两个点是1.5和2这两个阈限值。在接下来的实验中，将对这两个阈值分别进行对比试验。

SVM内部横向比较：

首先卡方检验量的阈值为1.5和2的情况下，分别得到每一个类别的特征为度数量如下表所示：

可以看到，

SVM、朴素贝叶斯、匹配分类横向比较。 图表

## 情感分析实验

### 情感分析算法实现

伪代码。

### 情感分析数据来源

情感分析的数据主要来自热点抽取止呕的文本，每一个热点均有40-100篇不等的相关文章。对其进行处理后，利于上节提出的分析算法，进行情感值的比较计算

### 情感分析实验结果呈现

情感分析的数据主要来自热点抽取止呕的文本，每一个热点均有40-100篇不等的相关文

# 总结及展望

## 总结

### 

## 工作不足及展望

### 

参考文献

[1]广西壮族自治区林业厅. 广西自然保护区[M].北京:中国林业出版社，1993:35-37.

[2]霍斯尼.谷物科学与工艺学原理[M].李庆龙，译.2版.北京：中国食品出版社，1989: 15-20.

[3]孙玉文. 汉语变调构词研究[D]. 北京：北京大学出版社，2000.

[4]赵耀东. 新时代的工业工程师[M/OL].台北：天下文化出版社, 1998[1998-09-26]. http://www.ie.nthu.edu.tw/info/ie ie.new.htm .

[5]全国信息与文献工作标准化技术委员会出版物格式分委员会. GB/T 12450 —2001 图书书名页 [S]. 北京: 中国标准出版社,2002.

[6]PEEBLES P Z, Jr. Probability, random variable, and random signal principles[M]. 4th ed. New York: McGraw Hill, 2001：100-110.

[7]韩吉人. 论职工教育的特点［G］// 中国职工教育研究 会. 职工教育研究论文集. 北京：人民教育出版社，1985: 90-99.

[8]赵颖力, 曹敏, 王琳.《化工学报》编辑部的人才建设[C] // 第 3 屆中国科技期刊青年编辑学术研讨会论文集. 北京：中国科学技术期刊编辑学会青年工作委员会, 2003：86-88 .

[9]张旭, 张通和, 易钟珍. 采用磁过滤MEVVA 源制类金刚石膜的研究[ J ]. 北京师范大学学报：自然科学版，2002，38（4）：478-481.

[10]周桂莲, 许育彬, 杨智全. 认清市场形势 化解“学报情结”：我国农业学报的现状与发展趋势分析[J]. 编辑学报, 2005, 17（3）：209-211.

[11]萧钰. 出版业信息化迈入快车道[EB/OL] .（2001-12-19） [2002-04-15]. http:∥www.creader.com/news/ 200112190019.htm.

[12]Online Computer Library Center, Inc. History of OCLC［EB/OL］. [2000-01-08]. http:∥www.oclc.org/about/history/default. htm.

[13]Nenad Medvidovic, Richard Taylor. A Classification and Comparison Framework for Software Architecture Description Languages[J] . IEEE Transactions on Software Engineering, 2000, 25（1）:70-93

**注意：**

**参考文献的排列按照学位论文中所引用的文献顺序排列，论文中参考文献引用需用上标。**

**文献数量合理，不太少也不滥用，文后列出的参考文献在正文中必须有对应的引用。**

**文献来源正宗权威，是学术文献，出典可查。**

作者简历

教育经历：

×××××××××

×××××××××

工作经历：

×××××××××

×××××××××

攻读学位期间发表的论文和完成的工作简历：

×××××××××

×××××××××

致谢

署名

于浙江大学软件学院

当前日期