**基于文本分类的金融舆情感知系统**

魏康健

目录

[一、 系统的需求规格说明书 2](#_Toc29951)

[2.1概述 2](#_Toc11837)

[2.1.1编写目的 2](#_Toc16869)

[2.1.2参考资料 2](#_Toc24051)

[2.1.3运行环境 2](#_Toc1322)

[2.2 功能性需求 2](#_Toc23573)

[2.2.1获取数据 2](#_Toc30099)

[2.2.2数据预处 2](#_Toc651)

[2.2.3构建文本分类模型 3](#_Toc29373)

[2.2.4进行评估 3](#_Toc12937)

[2.2.5可视化指数 3](#_Toc236)

[2.3 非功能性需求 3](#_Toc28572)

[2.3.1可靠性需求 3](#_Toc19564)

[2.3.2易用性需求 3](#_Toc22543)

[2.3.3安全性需求 4](#_Toc14560)

[2.3.4可维护性需求 4](#_Toc3437)

[2.3.5可扩充性需求 4](#_Toc25657)

[2.3.6性能需求 4](#_Toc30256)

[二、 概要设计 4](#_Toc550)

[三、 详细设计 5](#_Toc18916)

[4.1向量空间模型（VSM） 5](#_Toc23610)

[4.2 LDA 5](#_Toc9017)

[4.2.1LDA的生成过程 6](#_Toc25635)

[4.2.2LDA的整体流程 6](#_Toc28642)

[4.2.3LDA的学习过程 6](#_Toc28829)

[4.3短文本分类算法的具体实现 7](#_Toc5172)

[4.3.1：数据的爬取和预处理 7](#_Toc17165)

[4.3.2：文本分词 8](#_Toc2263)

[4.3.3 特征选取 8](#_Toc8937)

[4.3.7 SVM模型训练 11](#_Toc11030)

[四、 系统的开发环境配置 12](#_Toc1360)

[五、 系统的安装和使用说明书 12](#_Toc31885)

[六、 分类效果和舆情指数对比展示 14](#_Toc10886)

[七、 系统的先进性说明 16](#_Toc13626)

# 系统的需求规格说明书

## 2.1概述

### 2.1.1编写目的

本设计书简单阐明了系统的基本设计思想、基本功能、模块划分以及模块间接口，以便保证开发的质量、需求的完整性与可追溯性。保证需求分析人员、开发人员、测试人员更好地了解该系统的基本情况及各模块的详细功能，对需求达成共识。

### 2.1.2参考资料

自然语言处理基于java实现之基于VSM模型的信息检索程序——CSDN

### 2.1.3运行环境

Windows操作系统，JDK1.8及以上，Eclipse。

## 2.2 功能性需求

### 2.2.1获取数据

从股吧中爬取了大量股民对于股市的帖子的数据，从已经获取的数据中挑选出一定数量的样本数据，并人工进行类别标注（正向，中立，负向），建立训练样本集。

### 2.2.2数据预处

对挑选出的训练样本集进行数据预处理，将每个文本转换为词项的集合。

### 2.2.3构建文本分类模型

选用合适的文本表示模型完成训练样本集的文本表示，基于训练样本集来学习构建文本分类模型。本系统采用的是svm模型。

### 2.2.4进行评估

使用该分类模型对一个时间段内的每日股评数据进行情绪评估，将其分类为正向、中立、负向三个方面，并计算出每日金融舆情综合指数。

### 2.2.5可视化指数

利用数据处理工具将数据转化为不同特征对比图、十折验证图、优化前优化后图以及股票指数对比图来实现其可视化呈现每日金融舆情综合指数（与沪深300指数对比）。

## 2.3 非功能性需求

### 2.3.1可靠性需求

在规定的条件下，在规定的时间内，软件不引起系统失效的概率；在规定的时间周期内，在所述条件下程序执行所要求的功能的能力；其中的概率是系统输入和系统使用的函数，也是软件中存在的故障的函数，系统输入将确定是否会遇到已存在的故障（如果故障存在的话）。

### 2.3.2易用性需求

易用性实际上是软件的相当重要的属性， 体现的是这个程序是否好使。如果程序不好使，人们就会趋于不使用它，无论它的设计与实现是多么精致优雅。

### 2.3.3安全性需求

包括传输加密，存储加密，可破解性，以及各种未被授权的用户行为如何防范和控制，都是安全范畴，这里的安全不单针对外部普通用户，也针对内部不同级别的权限用户的的控制。小到如何防范和处理用户在输入框里输入特殊字符来获得设计者未曾预料的结果，大到防范外部黑客和内部内鬼。安全很多时候不单依赖技术实现，同时非常依赖相应的制度和审计。

### 2.3.4可维护性需求

系统的可维护性是衡量一个系统的可修复(恢复)性和可改进性的难易程度。所谓可修复性是指在系统发生故障后能够排除(或抑制)故障予以修复，并返回到原来正常运行状态的可能性。而可改进性则是系统具有接受对现有功能的改进，增加新功能的可能性。

### 2.3.5可扩充性需求

设计良好的代码允许更多的功能在必要时可以被插入到适当的位置中。这样做的目的的是为了应对未来可能需要进行的修改，而造成代码被过度工程化地开发。可扩展性可以通过软件框架来实现：动态加载的插件、顶端有抽象接口的认真设计的类层次结构、有用的回调函数构造以及功能很有逻辑并且可塑性很强的代码结构。

### 2.3.6性能需求

在软件响应速度、结果精度、运行时资源消耗量等方面的要求。经过对数据的不断优化和算法的改进，目前识别的准确率已经达到了80%。

# 概要设计

随着科技的发展，时代的进步，手机、平板电脑等移动设备越来越受人们的欢迎，同时随着移动互联网的快速发展，手机短信、微博、即时聊天信息、邮件和用户评论等以中文短文本形式表示的信息呈现出爆炸式的增长。研究者越来越关注如何在数量庞大的文本中挖掘出有用的信息，特别是对短文本的研究。我们设计的系统是针对股吧股民评论的金融舆情感知系统，股民在股吧中发布一些自己对于当前股情的看法，比如看好或看跌，我们负责采集股民发布的信息，然后对其进行必要的文本分类处理，从而判断股民的舆论倾向，这有助于我们对当前股票形式有一个大致的掌握，也能更好地揣测出当前的经济形势。

我们系统采用的文本处理方式是短文本处理，我们系统主要是通过股吧评论的标题进行判断是否看好当前股市。就目前而言，对于较长文本的信息处理，我们可以采用传统的长文本分类算法，比如Bayes、SVM、KNN、决策树等。然而这些方法在短文本处理时会存在一些问题。同样的，目前文本表示最常用的模型是向量空间模型（VSM），但是VSM是一种适合于大规模语料的文本表示模型，故效果也不是很理想。所以，我们重新考虑了一种方式，我们首先使用传统的VSM模型表示短文本，其中的特征选择是使用卡方检验完成的，同时使用LDA模型预测短文本隶属的“主题”，把主题特征加入到短文本中去，从而对短文本进行拓展，最后使用SVM分类器进行分类。

# 详细设计

## 4.1向量空间模型（VSM）

向量空间模型是哈佛大学的Ｇ.Salton提出的一个用于文本表示的统计模型。向量空间模型将一个文本表示成空间向量的形式，并以特征项作为文本表示的基本单位。基于VSM的文本表示可以描述为：，其中n是文本特征项总数，即向量空间的维数，表示第j个特征项在文本  中的权值（权值的大小反映了该特征项对于文本的重要程度）。

一个有效的分类特征项应该技能体现所属类别的内容，又能将该类别同其他类别相区分。因此在实际应用中，往往将TF和IDF的信息联合使用，即同时考虑TF和IDF的信息，称这种特征项的权值计算方法为TF \* IDF算法。我们系统采用的算法就是TF\*IDF算法：特征词频率TF指的是某个特征词t在一篇文档出现的次数。TF值越高，意味着特征词t对于该文档越重要。其中，N是文档集合中所有文档的数目，df是某个特征词的文档频率。特征词的IDF值越高，意味着该特征词对于文档的区别意义越大 。

## 4.2 LDA

LDA可以认为是两个贝叶斯化的一元语言模型的集成。LDA是一个文档主题生成模型，所谓生成模型，就是说，我们认为一篇文章的每个词都是通过“以一定概率选择了某个主题，并从这个主题中以一定概率选择某个词语”这样一个过程得到。文档到主题服从多项式分布，主题到词服从多项式分布。

LDA是一种非监督机器学习技术，可以用来识别大规模文档集或语料库中潜藏的主题信息。它采用了词袋的方法，这种方法将每一篇文档视为一个词频向量，从而将文本信息转化为了易于建模的数字信息。但是词袋方法没有考虑词与词之间的顺序，这简化了问题的复杂性，同时也为模型的改进提供了契机。每一篇文档代表了一些主题所构成的一个概率分布，而每一个主题又代表了很多单词所构成的一个概率分布。

### 4.2.1LDA的生成过程

对于语料库中的每篇文档，LDA定义了如下生成过程：

1.对每一篇文档，从主题分布中抽取一个主题；

2.从上述被抽到的主题所对应的单词分布中抽取一个单词；

3.重复上述过程直至遍历文档中的每一个单词。

语料库中的每一篇文档与K个主题的一个多项分布相对应（K的数值通过反复试验等方法事先给定），将该多项分布记为θ。每个主题又与词汇表中的V个单词的一个多项分布相对应，将这个多项分布记为φ。

### 4.2.2LDA的整体流程

先定义一些字母的含义：文档集合D，主题集合T

D中每个文档d看作一个单词序列<w1,w2,...,wn>，wi表示第i个单词，设d有n个单词。

D中涉及的所有不同单词组成一个大集合VOCABULARY（简称VOC），LDA以文档集合D作为输入，希望训练出的两个结果向量（设聚成k个topic，VOC中共包含m个词）：

对每个D中的文档d，对应到不同Topic的概率θd<pt1,...,ptk>，其中，pti表示d对应T中第i个topic的概率。计算方法是直观的，pti=nti/n，其中nti表示d中对应第i个topic的词的数目，n是d中所有词的总数。

对每个T中的topict，生成不同单词的概率φt<pw1,...,pwm>，其中，pwi表示t生成VOC中第i个单词的概率。计算方法同样很直观，pwi=Nwi/N，其中Nwi表示对应到topict的VOC中第i个单词的数目，N表示所有对应到topict的单词总数。

LDA的核心公式如下：

p(w|d)=p(w|t)\*p(t|d)

直观的看这个公式，就是以Topic作为中间层，可以通过当前的θd和φt给出了文档d中出现单词w的概率。其中p(t|d)利用θd计算得到，p(w|t)利用φt计算得到。

实际上，利用当前的θd和φt，我们可以为一个文档中的一个单词计算它对应任意一个Topic时的p(w|d)，然后根据这些结果来更新这个词应该对应的topic。然后，如果这个更新改变了这个单词所对应的Topic，就会反过来影响θd和φt。

### 4.2.3LDA的学习过程

LDA算法开始时，先随机地给θd和φt赋值（对所有的d和t）。然后上述过程不断重复，最终收敛到的结果就是LDA的输出。再详细说一下这个迭代的学习过程：

1.针对一个特定的文档ds中的第i单词wi，如果令该单词对应的topic为tj，可以把上述公式改写为：

pj(wi|ds)=p(wi|tj)\*p(tj|ds)

2.现在我们可以枚举T中的topic，得到所有的pj(wi|ds)，其中j取值1~k。然后可以根据这些概率值结果为ds中的第i个单词wi选择一个topic。最简单的想法是取令pj(wi|ds)最大的tj，即argmax[j]pj(wi|ds)

3.然后，如果ds中的第i个单词wi在这里选择了一个与原先不同的topic，就会对θd和φt有影响了。它们的影响又会反过来影响对上面提到的p(w|d)的计算。对D中所有的d中的所有w进行一次p(w|d)的计算并重新选择topic看作一次迭代。这样进行n次循环迭代之后，就会收敛到LDA所需要的结果了。

## **4.3短文本分类算法的具体实现**

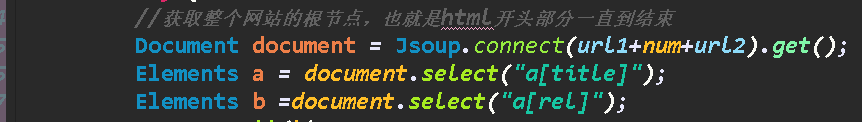
基于LDA特征扩展的短文本分类流程图如下图所示。



### 4.3.1：数据的爬取和预处理

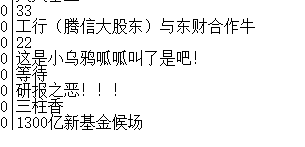
4.3.1.1：数据爬取：

在数据的爬取上选用网上的现有API可以方便的进行数据分页的爬取，我们选用Jsoup进行数据的爬取。获取的数据源是东方财富网。在数据的选取上，考虑到短文本的限制，我们只爬取标题，选取对应的标签，爬取大概8000条数据，核心代码如下。



4.3.2：数据的预处理

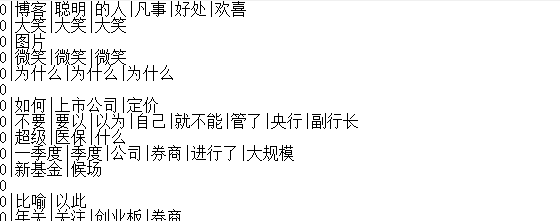
在爬取到了大量的数据后，我们发现爬取到的数据的质量有很大的区别。有些短文本仅仅包含两三个无用并且没有实际意义的词汇。这样的低质量数据对我们的SVM模型有很大的干扰，第一次因为没有对数据做正确的处理，导致我们的第一次分类准确率只有40%左右。在爬取到的数据中，有大量类似这样的无用干扰数据。



这种单纯的数字很难去判断，在数据预处理的时候就应该将其去掉，在去除了大量低质量的数据后在选择合适的算法和优化。

### 4.3.2：文本分词

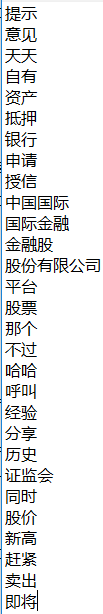
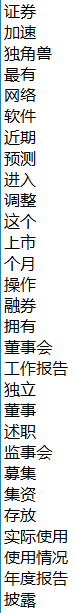
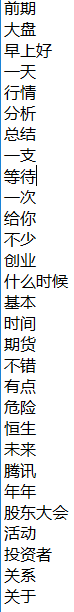
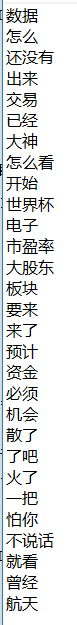
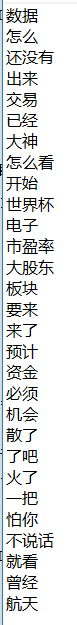
在进行特征选择之前，首先对训练集中的每篇文档进行分词，我们采用的分词工具是IKAnalyzer。IKAnalyzer是一个开源的，基于java语言开发的轻量级的中文分词工具包。我们选用此分词工具完成训练算法前的句子分词。分词后的结果如下。



### 4.3.3 特征选取

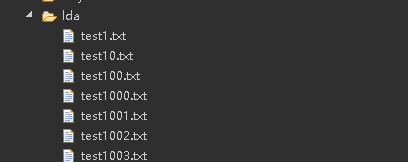
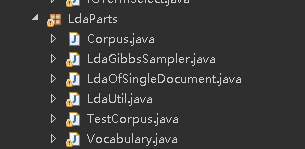
现有的特征抽取的方法很多，包括信息增益IG，局部文档频率和全局文档频率，对于文本分类选用局部DF或者IG都能达到良好的效果。传统的VSM向量空间都是根据这里的特征选取的特征词进行文本表示。

我们的算法第一步根据IG或者局部DF先完成普通的特征选取。然后通过训练得到基于LDA隐主题模型选取的，隐主题对应的高比重的单词进行特征加强。即对于原来特征选择中没有出现的单词，但在LDA主题模型中出现的单词将其加入。完成LDA拓展的文本特征选取。获得的部分特征词如下：

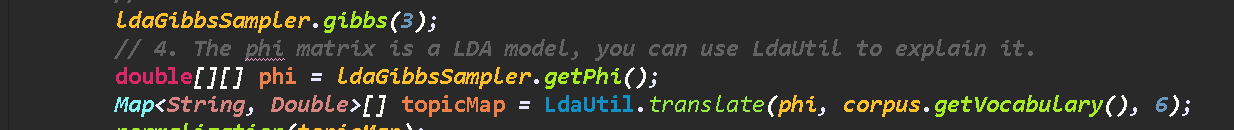


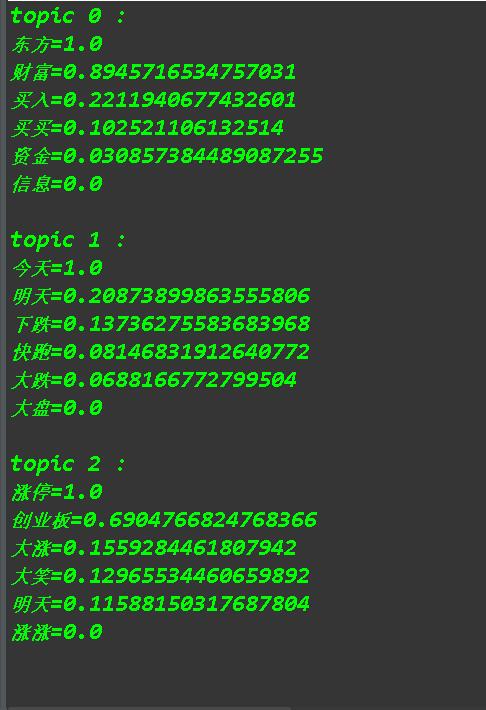
**4.3.4：LDA隐主题的抽取**

为了完成LDA拓展的特征选取，我们要先训练LDA得到隐藏的主题。找到每个主题下的对应关键词项。LDA的模块使用的是GitHub上成熟的LdaGibbsSampler的API模块。但为了得到短文本的LDA主题模型，我们需要将这上千条数据划分为上千个单独的短文本文档。具体实现在下方LdaOfSingleDocument中。



将数据处理成了lda可以处理的格式后，我们设定预期的主题是3个，分别对应股市的乐观，悲观，和中立三种。在进行LDA主题的处理后，我们得到下面的隐主题。





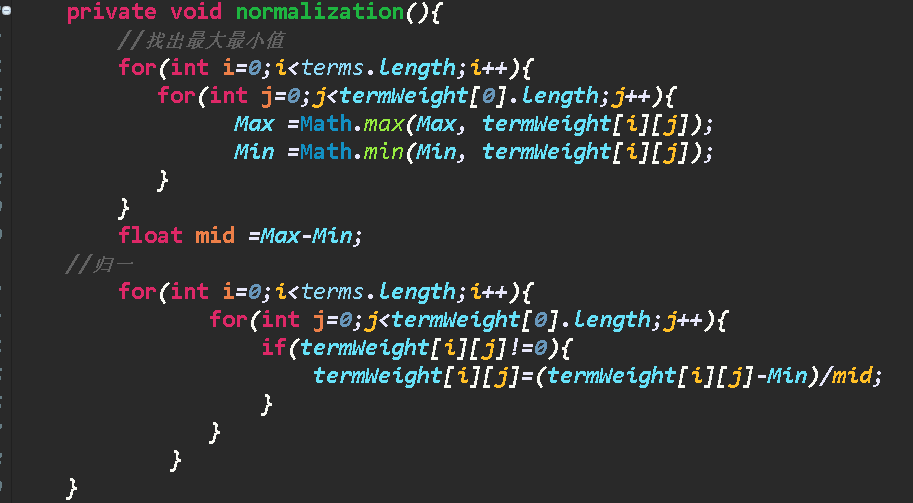
可以清晰看出，隐主题的划分对应的关键字信息，恰好对应的股市舆情为无关（topic0）,乐观（topic1）, 悲观（topic2）。即LDA隐主题的特征抽取给我们提供了关键的数据。

**4.3.5：基于LDA特征的拓展**

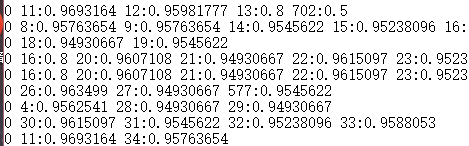
在得到了隐主题对应的关键字后，基于之前我们已经通过局部DF或者IG获得的特征词，进行拓展，对于原来没有获取的特征词而在LDA主题模型中出现了的，将其加入。完成LDA的拓展的特征词的选取。

**4.3.6：特征的文本表示和数据归一化**

得到特征词后，选用tf\*idf进行文本表示， tf\*idf是常用的反应本文本中词汇出现个数和其他文本中没有出现个数的一种很好的表示方法。因为LDA计算的关键字的频率是处于0到1之间的，而原本tf\*idf的文本表示结果是远远小的多的。在这里进行归一化，将原本小的多的数据放缩到0和1之间，从而方便应用LDA选取词项的权重。归一化过程如下：



这样将原本DF选取的特征词和LDA隐主题选取的特征词都进行了0到1之间的表示，输出的文本表示样例如下：



这样完成了LDA和原有特征词在VSM空间向量下的文本表示。

### 4.3.7 SVM模型训练

有了文本表示的输出文件，我们就可以进行SVM的训练，这里没什么新的地方，就是用老师提供的LIBSVM进行模型的训练。值得一提的是这里设置的松弛变量的设置来避免一些可以抛去的和SVM分类平面较远的坏点。还有核函数的选择。这里都是没什么可以改进的地方，已经很成熟了。

# 系统的开发环境配置

开发环境：WIN10 JDK1.8以上

编译器：Eclipse

额外必须JAR包：



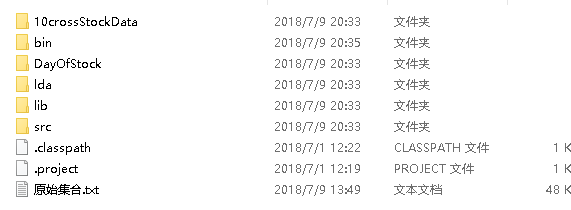
在打包的压缩包中已经给出

# 系统的安装和使用说明书

打开压缩包下的TextCategorizationForStock文件夹的.project导入项目

文件中包含所有的数据，包括原始数据，十折交叉验证的数据

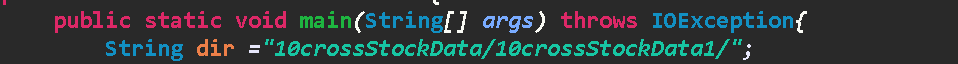
和爬取用来和股市指数对比的6.1到6.30每天的股市评论



导入项目后，在对应的TrainAndCategorization包下的两个JAVA文件中分别进行训练集的模型训练。和用模型对测试集进行测试。

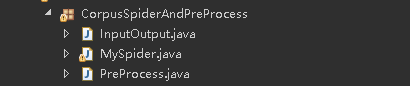


数据集合的路径已经包含正所给的项目中。具体实现细节已经封装好，只需改每个java文件中，String类型的dir中的最后的数字，即可对不同的训练集和测试集进行训练和测试。



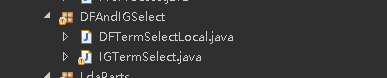
**系统其他部分说明：**

CorpusSpiderAndPreProcess：



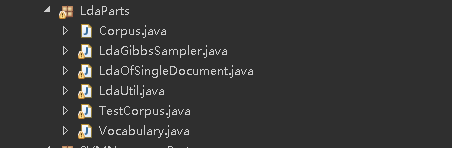
工具类的定义，和爬虫的编写，以及用IKAnlsaly进行文本分词的预处理

DFAndIGSelect：



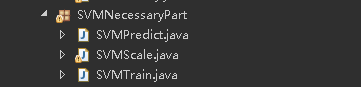
两种不同的文本表示方式，用来优化和对比

LdaParts



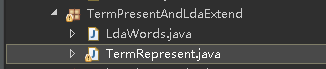
LDA隐主题抽取的必要部件，系统的先进性核心部分，包括对原有文本的分成单独文本，和对隐藏主题的抽取，以及设置LDA隐主题的个数和每个主题选取几个关键字。

SVMNecessaryPart



SVM模型训练必要文件，没有进行优化。

TermPresentAndLdaExtend

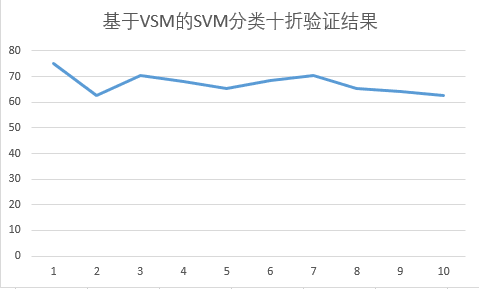


LDA可拓展的核心部分，系统先进性的核心核心部分，包括数据的归一化和LDA拓展隐主题的特征词项目的加入。

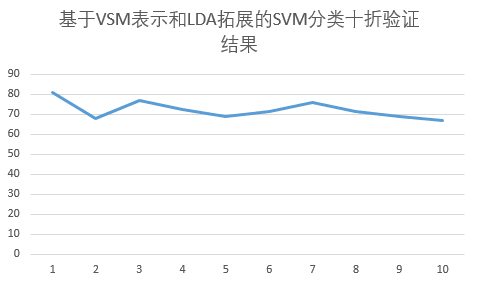
# 分类效果和舆情指数对比展示

**经过LDA拓展优化后的模型，训练集自身测试准确率大概在90左右。十折交叉验证和对外部数据的测试准确率大概在70-75左右。**

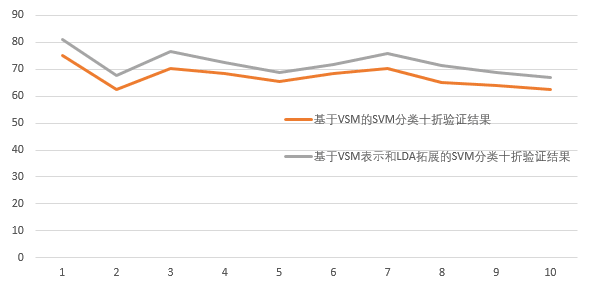
**普通DF特征选择SVM分类十折验证**



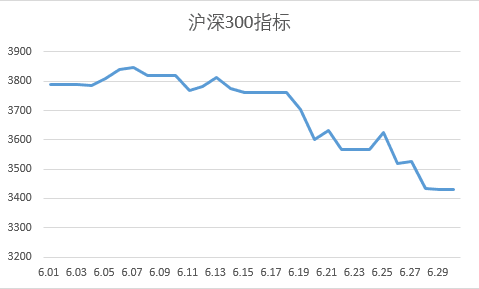
**基于LDA拓展的加入隐主题的SVM分类十折验证**

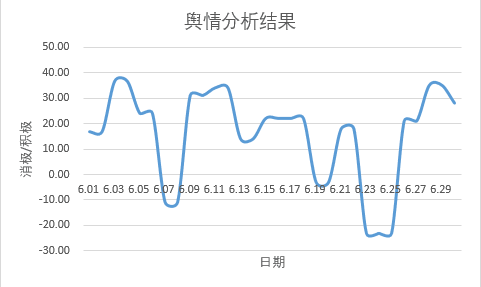


**优化前和优化后准确率对比图**

****

**用所得模型对6月份股吧数据爬取，得到舆情指数和大盘指数的对比图**

****

****

**注：舆情指数计算公式：（积极的评论数-消极的评论数）/（积极的评论数+消极的评论数）**

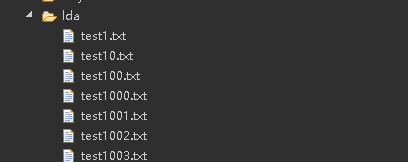
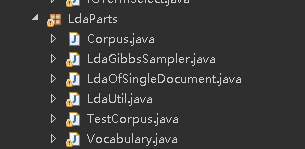
**可以看出，情感指数和股票趋势大致符合，情感指数和股票指数大致变化相同，当股市下跌，情感指数降为负值。股市行情回升，情感指数业迅速上升为正。在6.5-6.9和6.19到6.27之间都较好的拟合。**

# 系统的先进性说明

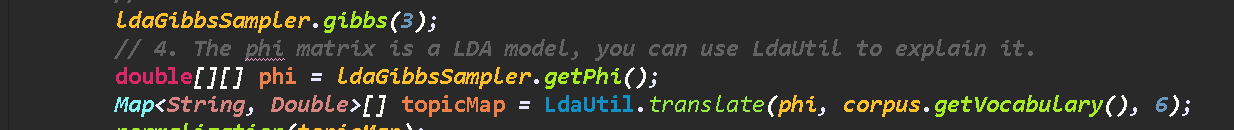
系统的先进性按其重要程度排序体现在下面几个方面：

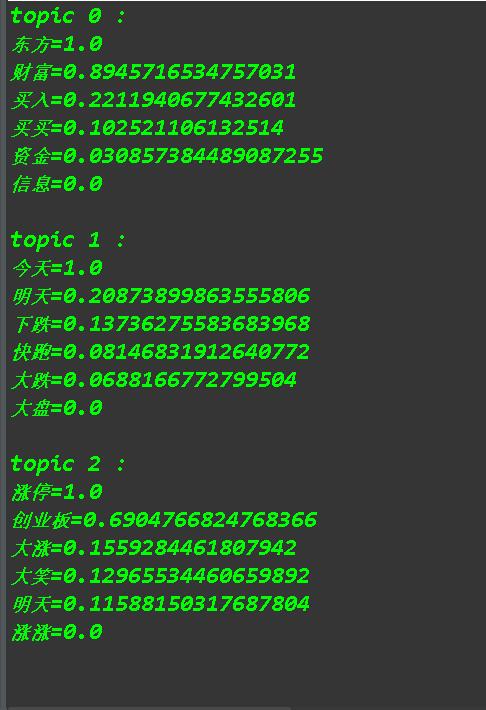
**1：引入LDA概率主题模型，得到隐主题对应的关键词**

为了完成LDA拓展的特征选取，我们要先训练LDA得到隐藏的主题。找到每个主题下的对应关键词项。LDA的模块使用的是GitHub上成熟的LdaGibbsSampler的API模块。但为了得到短文本的LDA主题模型，我们需要将这上千条数据划分为上千个单独的短文本文档。具体实现在下方LdaOfSingleDocument中。



将数据处理成了lda可以处理的格式后，我们设定预期的主题是3个，分别对应股市的乐观，悲观，和中立三种。在进行LDA主题的处理后，我们得到下面的隐主题。



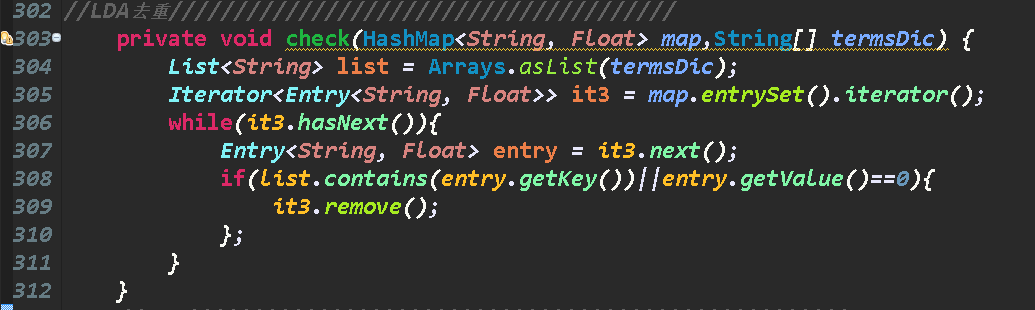


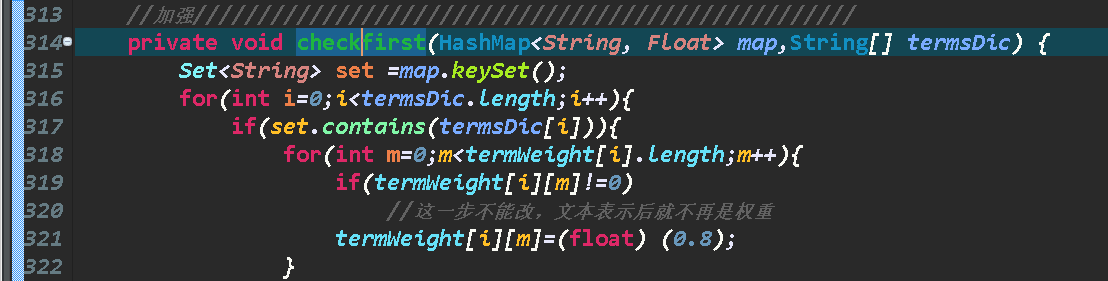
可以清晰看出，隐主题的划分对应的关键字信息，恰好对应的股市舆情为无关（topic0）,乐观（topic1）, 悲观（topic2）。即LDA隐主题的特征抽取给我们提供了关键的数据。

**2：用LDA得到的特征词拓展和加强特征词典**

这一块具体实现在

/TextCategorizationForStock/src/TermPresentAndLdaExtend/TermRepresent.java 这个文件下。通过函数check()完成LDA的去重，即如果LDA的词项在原有的DF抽取的特征词词典中，则去掉LDA的此项。通过函数checkfirst（）完成LDA的加强。如果LDA抽取的隐主题应该被添加，则不再用tf\*idf进行文本表示的计算，直接赋予一相等值，即使具有LDA词项的文本的文本表示值到距离SVM分类平面的距离人为缩短。完成优化。具体两个函数实现细节如下：



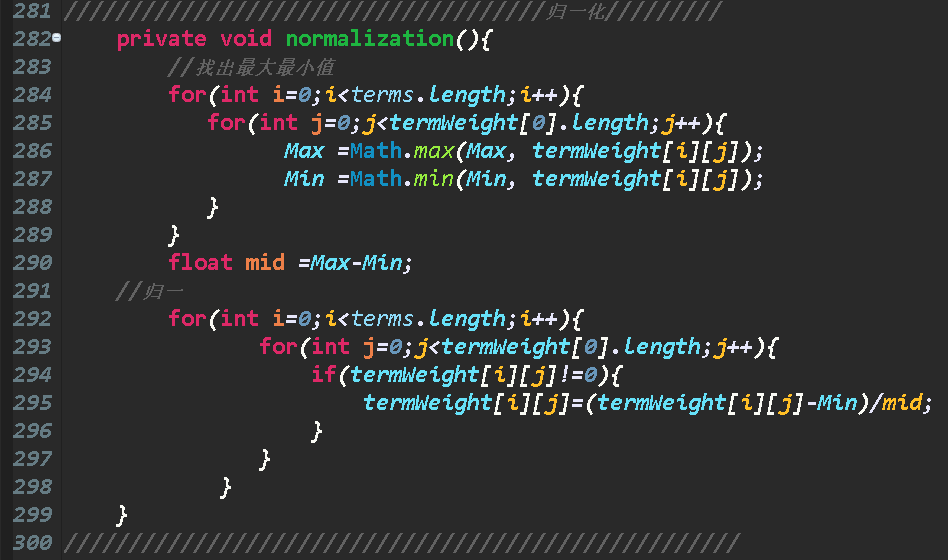


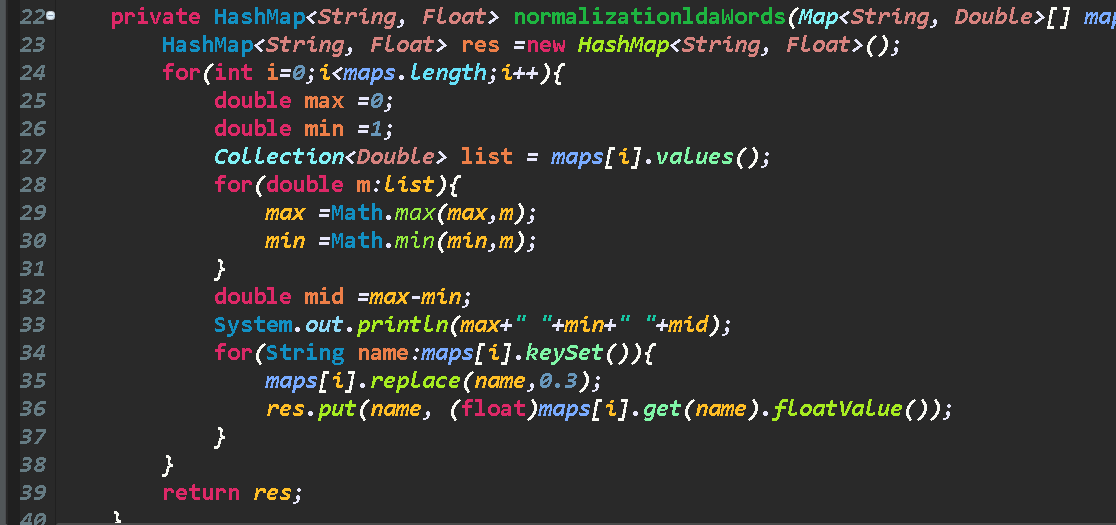
**3：数据归一化精确表示特征词**

为了使得LDA获得隐主题对应的关键词的权重能被用于VSM的文本表示，将原本文本特征表示e\*10^-14这样量级的数据放缩到0和1之间，方便LDA的拓展。具体实现函数如下。这一块具体实现在

/TextCategorizationForStock/src/TermPresentAndLdaExtend/TermRepresent.：

和/TextCategorizationForStock/src/TermPresentAndLdaExtend/LdaWords.java：



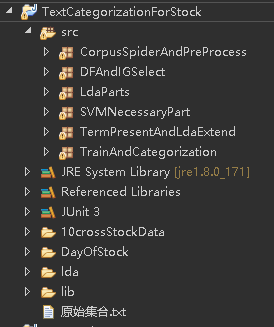


**4：数据的预处理和平衡数据集**

在爬取到了大量的数据后，我们发现爬取到的数据的质量有很大的区别。有些短文本仅仅包含两三个无用并且没有实际意义的词汇。这样的低质量数据对我们的SVM模型有很大的干扰，第一次因为没有对数据做正确的处理，导致我们的第一次分类准确率只有40%左右。在爬取到的数据中，有大量类似单纯数字和无用字符的干扰数据。这种单纯的数字很难去判断，在数据预处理的时候就应该将其去掉，在去除了大量低质量的数据后在选择合适的算法和优化。

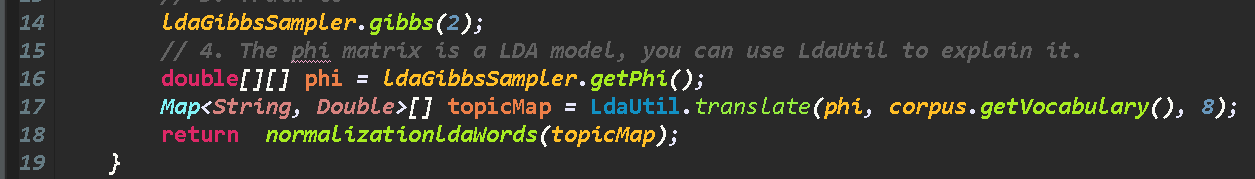
在选取的数据集打完标签后，处于0，1，2状态的标签样本数量不一样，这种数据集的不平衡性，对模型的分类结果是有影响的，为了使得数据集相对平衡，我们爬取了额外的大量数据，来选取样本数较少的类别的数据。保持数据集的平衡性，提高模型的准确率。

**5：良好的系统松耦合，高度的封装和较好的可拓展性**



如上图，系统将不同模块功能的组件组合在一起。如下图。每个模块均可以单独完成本部分的工作，如爬虫部分。分词。特征选取。SVM组件。文本表示和最后训练模型和分类模块均做了抽象处理。封装在不同的包中。可以直观可见处理的过程。对于处理的主函数模块，可直接选取数据集进行训练，其他模块对其透明，提高了可用性。系统暴露出主要public函数方法，可以整体封装为jar包，做为短文本分类的组件使用。

良好的可拓展性体现在，系统对于影响分类结果的可变参数是支持修改的，比如影响分类效果的LDA隐主题个数的设定，和每个主题包含词数的定义，开放修改，如下图。通过调参完成对不同短文本的适应。



**6：SVM分类结果的详细表示，输出预测的文本对应的标签和分类错的文本**

原本SVM的SvmPredict的组件只能输出分类的正确率，对于出错的分类文本不知道是那里出错，通过在

/TextCategorizationForStock/src/SVMNecessaryPart/SVMPredict.java中增加分类集合结果的输出，从而不仅知道正确率，还能对应分类错的文本进行查找和方便优化算法。

改进后的输出如下：

