哈爾濱工業大學

毕业设计（论文）

题　　目　面向不均匀类别的文本分类系统设计与实现

专　　业　计算机科学与技术

学　　号　11103101**

学　　生　徐*

指 导 教 师　***

答 辩 日 期　2015.06.30

摘　要

本文以设计并实现面向不均匀类别的文本分类系统为目的，主要调研了传统文本分类模型和基于深度学习的分类方法，包括 Bag-Of-Words 模型，Ngram（unigram 和 unigram 与 bigram 组合）表示效果，布尔、归一化 TFIDF、NB 特征表示，SVM 和 Boosting 方法，以及卷积神经网络模型。同时关注数据不均衡下的处理方法，主要涉及过采样、下采样，调整分类器参数及 SMOTE 方法。在足够的理论基础上，结合开源资源完成实验设计与代码编写。在英文 IMDB 数据集和搜狗 IT 和科技类别数据集上完成均匀和不均匀数据的分类效果测试，在 IMDB 数据集上比较了过采样、下采样，SVM 调参与 SMOTE 方法的实际效果。最后通过对实验结果的分析，从中总结出面向不均匀类别的文本分类系统的设计方案。设计 1 使用 BOOSTING 方法，使用文本的 unigram 布尔特征，对不均衡数据集做过采样处理，构建出稳定适应性强的分类系统。设计 2 使用 unigram 与 bigram 组合及归一化 TFIDF 特征，以 SVM 作为分类方法，通过调整 SVM 参数完成特定数据集上的分类系统构建。实验的实现代码已公布在互联网上，主要包含各方法的特征抽取代码和工作流脚本。

关键词：文本分类；数据集不均衡；数据偏置；

Abstract

With the purpose to design and build a text classification system facing to unbalanced dataset , this paper not only talks about the text classification methods of classical and satte-of-the-art deep learning , but also focus on data-skewd problem .The Bag-Of-Words model , text representation by bool or normalized TFIDF or NB feature , classifier including SVM and BOOSTING , and Convolutional Neural Network model has been surveyed . Absolutely , we also focus on mothods for handling data skewing problem . Over-sampling , under-sampling , parameter tuning and SMOTE mothods are all applied on experiment. After finishing readying the sufficient theory basis , experiment design and coding has been done with the help of open source resources . we do the expiriments on the IMDB dataset of english and Chinese Sogou IT-Tech dataset to get the results about balance and skwed datas . Ohterwise , we test over-sampling , under-samping , parameter tuning for SVM and SMOTE mothods on IMDB unbalanced dataset and compare the results . By analizing the results , we design 2 workflows . The first is to get a robust and not bad classify system by combining unigram , bool feature , BOOSTING and over-sampling . The onther aims to obtain a higher results and uses SVM , unigram and bigram , as well as regularized TFIDF where SVM should be parameter tuning .The experiment code is available on Internet which including feature abstracting and workflow shells .

Keywords: text classification , unbalaned dataset , skewed dataset

目录

第 1 章 绪　论

1.1 课题背景及研究的目的和意义

无时无刻，我们都在采集、存储数据。这些数据可能来自互联网终端节点上的一个人，也可能源于监测地震信息的传感器群。我们处在信息爆炸的时代，需要一定的自动化手段对获取的数据进行操作。基于机器学习方法的文本分类就是其中的一种。

文本分类是在现实生活中经常要面对的问题。譬如我们浏览新闻站点，顶部导航栏内通常包含新闻的类别，而主体内容则可能混合显示各个类别的新闻。当我们选中并点击某一个类别链接，往往会跳转到该新闻类别的页面，显示仅属于该子类别的新闻。当下社交网络流行，热门事件常常通过社交网络迅速传播。当我们通过社交网络看到某个热点新闻时，往往也会通过社交媒体表达自己的态度，这时一些统计机构往往能够获取到社交媒体使用者对该新闻的表态，并对这些可能包含喜悦、愤怒词语的文本进行分类，得出宏观下普遍大众对该热点新闻的态度趋向。由上述生活中常见的一些例子，管中窥豹，可知文本分类问题在日常生活中是普遍存在的。

实际生活中的数据，往往都是类别不均衡的。例如对于普通人而言，收到的垃圾邮件与正常邮件的数量是不均等的，且往往相差很大。对于微博头条，娱乐信息往往远多于科技信息。这些类别不均衡的数据，给我们自动化文本分类带来了麻烦。

机器学习方法往往面向的都是均匀类别的数据，在不均匀类别上一般效果不佳。面向不均匀类别的文本分类系统设计与实现，就是希望通过调研目前的文本分类方法以及在不均衡数据集下的处理方式，总结出可应用于实际问题中的、效果稳定且优秀的文本分类方法。

文本分类在过去的时间里已有大量的、充分的研究，并且这些方法已经取得了良好的效果。而数据不均匀问题也很早就被研究者们重视，已有一些较为通用的处理办法。由于本人知识和能力的限制，并不能在此重要且成熟的领域做出有效的创新。研究该课题，主要目的还是希望对文本分类问题能有较为全面的理解，并在此基础上将面向不均匀数据的多种处理手段和不同文本分类方法组合，通过较为充分的实验，找到可应用于实际的面向不均匀类别的文本分类方法。在此之外，也希望尝试学习与深度学习相关的前沿方法并将其应用于文本分类实验中。

1.2 相关研究工作

在[1]中，Zhang 研究了从医学文献中抽取蛋白质名词这一问题。在该问题中，蛋白质名词在训练集中仅占 4%，Zhang 通过对非蛋白质名词下采样得到正负例相对均衡的数据，然后以此作为 K 近邻方法的训练集，完成不均匀数据下的信息抽取。下采样的抽取方法可以是随机的，也可能是经过特殊规则选择部分数据进行采样。Zhang 是[1]中尝试了 3 种下采样方法来选取特定的实例，一种是选择负例实例中到部分正例数据的距离和最小的数据，一种是选择离全部正例数据距离和最小的数据，最后是在每个正例周围选取最近的 K 个负例，然而最终实验结果表明这些方法并没有比随机选取有更好的效果。

Chawla 从特征向量的角度，提出了 SMOTE(Synthetic Minority Over-sampling Technique)方法[2]。通过在某个特征向量和其相邻特征向量间构建一个新特征向量，SMOTE 实现了有别于传统有放回随机抽样的过采样方法。Chawla 在[2]中通过大量实验说明该方法有助于提高在少量类别数据上的分类准确率。在实验中，Chawla 使用了 ROC(Receiver Operating Character)评价方法，并说明该评价方法适合于不均匀数据集下分类效果的评估。

Akbani 在[3]中以 SVM 分类器为基础，通过调节分类器对不同类别的权值比重，同时结合下采样和 SMOTE 方法[2]构建均匀训练集，并以此为基础训练得到 SVM 分类模型。[3]中通过实验表明该组合方法取得了较任何单一方法更好的效果。

1.3 本文的主要研究内容

本文主要以现有方法调研和实验验证为主。通过尽可能多的调研方法并完成实验效果对比，总结出面向不均匀类别的文本分类系统的设计与实现方法。

方法调研上主要学习传统文本分类模型和基于深度学习的分类方法，包括 Bag-Of-Words 模型，Ngram（unigram 和 unigram 与 bigram 组合）表示效果，布尔、归一化 TFIDF、NB 特征表示，SVM 和 Boosting 方法，以及卷积神经网络模型。同时关注数据不均衡下的处理方法，主要涉及过采样、下采样，调整分类器参数及 SMOTE 方法。通过调研这些方法，从中积累到足够的理论知识。

实验验证上使用开源工具，编写代完成实验设计。在英文 IMDB 数据集和搜狗 IT 和科技类别数据集上完成均匀和不均匀数据的分类效果测试，在 IMDB 数据集上比较了过采样、下采样，SVM 调参与 SMOTE 方法的实际效果。最后通过对实验结果的分析，从中总结出面向不均匀类别的文本分类系统的设计方案。

第 2 章 文本分类问题调研

2.1 引言

文本分类问题是自然语言处理中的一个基本问题，通常包含主题文本分类，情感分类，问题分类等多个子问题。主题分类可以帮助我们分类新闻，方便阅读；情感分类可使我们快捷廉价地获取群体情感趋向；问题分类则能够方便问答系统处理问题。

由于文本分类与我们的日常生活息息相关，人们很早就将注意力投入到该问题上来。通过对该问题，学者们提出了经典的 bag-of-words（中文可称做“词袋”模型，通常取各英文首字母，记为 BOW）方法。在随后的研究中研究者们尝试不断改进该方法，催生了一系列该模型的变体。基于 BOW 方法的模型，配合朴素贝叶斯分类、支持向量机等分类方法，在文本分类上取得了非常好的效果。我们在这里将基于 BOW 模型的方法称作传统文本分类方法。

近年来，一度销声匿迹、无人问津的神经网络方法，经过 LeCun 等学者的改进，并伴随着计算机运算能力的提高、GPU 并行计算流行以及大数据“云”时代的到来，卷土重来，并大有一统天下之势。继深度神经网络在计算机视觉领域取得突破性进展[4]后，深度学习又攻下语音识别[5]这一领域。大批学者开始窥视仍然在使用传统机器学习方法和结构化特征的自然语言处理领域，希望使用深度学习的方法，在该领域同样做出突破。在这样的背景下，使用深度学习方法解决文本分类这一基础问题受到了学者们的重视。

本章将首先介绍传统 BOW 模型及一些变体，以及在该模型之上的广泛应用的分类方法。最后再尝试浅陋地总结当下大批学者使用深度学习方法在文本分类问题上的积极尝试。

2.2 BOW 模型及相关变体

经典的文本分类任务往往包含文本表示和分类器设计两个部分。BOW 模型完成了文本表示的功能。本节将介绍该模型如何将以自然语言形式存在的文本表示为机器能够识别的数值。

2.2.1 向量空间模型及 BOW 模型

首先需要介绍的是向量空间模型[6]417-419，传统的文本表示方法即建立在该模

型之上。向量空间模型(vector space model , VSM)是一种将文档表示成为空间向量的文本表示方法。以下对其做出较为详细的定义。首先定义文档的项(term)，这将被作为文档表示的最小单位。根据需求，项可以是文档中的单词、词组、片段或者是主题等。项将被映射为 VSM 的基。将一组文本中的所有相异项放在一起构成一个向量（可以理解为有序集合）,在本文中将该向量称为特征项向量，向量中的每一维被称作特征项(feature term)。假设特征项向量的维度为 N，我们以每个特征项为 VSM 的基，N 维特征项向量就可看作空间中的 N 维坐标系。我们指定每个文档在每个特征项上的值，就能将文档映射为 N 维空间下的向量（或者点）。本文中将文档在特征项上的值称作特征值。特征值可以是该特征项在该文档中出现的频数（TF），或者是标识该特征项是否在该文档出现的布尔表示。将特征值按对应特征项的顺序排列后即构成了 N 维数值向量，本文中将其称作特征值向量。以特征项向量为基底，通过每个文档的特征值向量将文档映射到空间中。这就是向量空间模型表示。

当我们把上述向量空间模型中的项取为单词，将特征值设为 TF 表示，该模型即被称作 BOW 模型，或词袋模型。词袋模型忽略了构成文本的单词的顺序，而只关注于单词出现的频次。

2.2.2 BOW 模型变体

词袋模型的变体体现在两个方面，一个是特征项的选取，一个是特征值的选择。

2.2.2.1 基于改变特征项选择的 BOW 模型变体

前面提到，传统的 BOW 模型将单词(或者称作 unigram)作为特征项，一般来说这已经能够较为充分的保留原始文本的信息，并为后续分类器的分类提供足够区分度。

但是，以 unigram 作为特征项也存在一定的缺陷。首先，词袋模型本身决定了单词间的顺序被忽略，这在一定程度上削弱了其文本原始信息的保留能力。一种可能的解决方法是使用更高阶的 Ngram（这里指 unigram 之上的 bigram，trigram 等）。确切的说，使用 unigram 加上 bigram 甚至加上 trigram 来增强信息的表达，在一个局部内保证单词的有序性。举例来说明使用高阶 Ngram 模型有效的一种可能理由[7]。假设在情感分类任务中，一个文本中有一句话为 *I think it is NOT BAD but VERY GOOD*，如果仅使用单词作为特征，那么“NOT”和“BAD”，“VERY”和“GOOD”将会被分开，并由于 BOW 忽略位置的属性，到分类器层次，这四个纯粹无序的单

词可能有不同的组合，且不同的组合可能表达完全不同的极性，导致原始文本的信息丢失。但如果我们加入 bigram 的特征，那么“NOT BAD”和”VERY GOOD”就能够得到保留，原始文本中的信息得以传递到分类器层次，分类器就存在可能学习到恰当的参数对文本进行合适的分类。这便是使用高阶 Ngram 作为特征项在保留原始语义上的优势。

由 unigram 向高阶 Ngram 的过渡非常自然，但是多年来高阶 Ngram 仍然很少被选择作为特征项。最根本的原因是加入高阶 Ngram 表示后，特征向量的维度将变得难以接受。举例来说，假设我们原始有 N 个单词，加入 bigram 后，极端情况下可能增加 N*N 个特征项。这种成平方关系增长的特征项数量让后续分类器难以处理，特别是在过去硬件受限的年代。同时，过多的特征项，对于一些分类器来说也意味着需要学习更多的参数，在数据量不扩增的情况下，往往会造成过拟合（Overfitting，指分类器过度拟合训练集而缺乏对训练集之外数据的泛化能力，导致实际分类效果不佳）的情况。最后，加入高阶 Ngram 后会使得特征向量更加稀疏。

在需要保留更多信息和保持文档向量维度可控这二者之间，人们不得做出权衡(Trade-Off)，过去的一段时间里人们仍普遍使用 unigram，但尝试高级 Ngram 未尝不是一个提高分类效果的尝试。

即使使用 unigram 作为特征项，特征向量的维度同样很高（往往能达到上万量级）。面对该问题，人们从分别从两个方向上去尝试解决。

一种比较直观的方法是对原始的特征项做一些筛选或映射。例如使用提取词干、大小写归一化的方法来减少英语中因为词形变化带来的特征项数量成倍增长的问题，这可以看做是一种映射方法。使用基于 IG（Information Gain，信息增益）、CHI（卡方统计量）等方法对单词对于文类结果的影响做出排序然后仅取前 K 个作为 VSM 中的基，这是基于特征项筛选的方法。这些方法往往可以合并使用。

另一方面，人们试图跳出以原始文本中的单词作为项的固有思维，找到其他的特征项。一个应用很广的尝试是使用语义特征来表示文本。所谓语义特征项，是在词的基础上做出的一定归并——即将具有一定联系的一批词聚合到一起作为一个特征项。语义特征项仅是本文中的称呼，并不具有通用性。使用语义特征项作为文本表示的方法减少了特征向量维度，且当语义抽象足够精确时同样能够表达原始文本信息。著名的 Latent Semantic Analysis(LSA)[8] 和 Latent Dirichlet Allocation(LDA)[9]即可认为是这种方法的代表。在[10]中 Lebret 等提出的基于 word embedding 的 bag of semantic concept 特征表示(后文称作 Bag-Of-SemanticConcept 方法，或简称为 Semantic-Concept)。这可以认为是将深度学习的成果用于传统文本

分类方法的尝试。以下将较为详细的论述该方法。

Bag-Of-SemanticConcept 方法的根本是 distributed representation[11,12]，即将单词表示为向量的方法。该方法的具体做法未调研，可以使用 word2vec 工具[1]完成这项工作。对于 unigram，其向量表示可直接取 word2vec 的结果，对于高阶 Ngram，文中采用了将组成高阶 Ngram 的所有 unigram 向量相加求均值的方式，即 $V_{\text{Ngram}} = \frac{1}{n}\sum_1^n V_{unigram_i}$。文中举了一个示例来表达该方式的合理性：V(“king”) – V(“man”) + V(“woman”) ≈ V(“queue”) 。该例子说明基于 word embedding 的几何运算一定程度上是能够保持其语义，其结果是有意义的。

使用 word2vec 工具完成 Ngram 向量化后，将这些代表 Ngram 的向量通过 K-Means 聚类，这些类别即是 semantic concept 。

特别地，对应于 semantic concept 特征项表示，文中提及了两种计算特征值的方法。一种方法是将每个 semantic concept 中包含的所有 Ngram 的频次之和作为特征值，另一种是使用后文中将提及的 NB 特征，即 log-count ratio。可以认为 log-count ratio 表示了每个特征项划分文档类别的能力，将其值记为 r。该方法取一个 semantic concept 类中包含的所有 Ngram 对应的 r 值中绝对值最大的 r 作为该类的特征值。

2.2.2.2 基于改变特征值计算方法的 BOW 模型变体

改变特征值计算方法非常简单易行。最常用的一种特征值表示是TF · IDF。TF 表示项在一个文档中出现的频次，而 IDF 是 Inverse Document Frequency 的缩写，中文名称作反文档频率，计算方法是：

$$\text{Idf} = \log\left(\frac{N}{n}\right)$$

其中 N 表示总文档数，n 表示包含该词的文档数。通常文档总数一定，故包含该词的文档越多，Idf 值越小，反之包含的文档数越少，Idf 值越大。在一定的程度上，这种表示与我们的常识是相符的。举例来说，中文中常见的一些词语如“的”,“我”等，往往在每篇文章、各个类别中都出现，这样的词语对于我们的分类问题显然是没有帮助的，我们应该减少它的比重；反过来，一些与类别极其相关的词语，如医学领域类别下的“维生素 B6”,“嘌呤”等词语（组），往往只出现在很少的文档中，但这些词语无疑对于类别的分类非常有帮助，故增大这些特征项的特征值是有道理的。同时也可以看到，TF 与 IDF 之间也有着相互调和的作用。在一个文档中出

[1] https://code.google.com/p/word2vec/

现次数较多的词，TF 能够给他们足够大的值表示；同时，对于被 TF 规则忽略的出现次数较少但对分类有效的词，IDF 给予其特征值足够的提升。TF·IDF 这种特征值表示方法，在信息检索中应用广泛，是久经验证的表示方法。不过，使用基于频次统计的特征表示方法，不可避免的引入了文本长度这一影响因子。在不同的文本中，同一个词的 TF*IDF 值的大小，不仅与其在文本中出现的概率有关，还很大程度受到文本长度的影响。以绝对 TF*IDF 作为特征值，如果目标的分类与文本长度无关，那么该特征值表示就会稍有不妥。例如对体育新闻的分类，一段体育短讯可能仅包含“进球”这一词语 2 次，而一篇关于某足球明星的长娱乐新闻则可能同样包含“进球”两次。这这种情况下显然需要考虑文本长度的影响。一种解决方法是使用归一化的 TF·IDF 特征表示，计算公式是：

$$\mathrm{TFIDF}_{\mathrm{norm}} = \frac{TF_i * IDF_i}{\sum_j TF_j * IDF_j}$$

通过求取一个相对值，较大程度上减少了文本长度对于分类结果的影响。

除了 TF·IDF 这一种表达词频和反文档频率相互调和的扩展外，还有一种仅仅表示该词是否在文档中出现的布尔特征值表示比较常用。布尔特征的优点是非常简单直观，且在一些分类器下表现并不显著弱于相对复杂的 TF·IDF 表示

上一节中，提及了使用 bag of semantic concept 来做特征项时可是选择使用 NB 特征。在[7]中，Wang 具体阐述了 NB 特征的定义及求法。首先，我们要把所有文档表示为布尔特征向量，将其记为$f^{(i)}$，i表示文档集中某个文档的索引。同时我们定义正负例文档标记$y^{(i)} \in [1, -1]$。在此基础上，定义$p = \alpha + \sum_{i:y^{(i)}=1} f^{(i)}$，表示将文档集中所有正例文档的特征向量相加，再加上一个平滑因子 α；类似定义负例文档的特征向量计算：$q = \alpha + \sum_{i:y^{(i)}=-1} f^{(i)}$；最后，我们定义$r = \log(\frac{p/\|p\|_1}{q/\|q\|_1})$，即将向量$p$和$q$使用一范数归一化后相除再取对数，其结果$r$被称作 log-count ratio，即所谓的 NB 特征。最后，使用r更新每个特征向量$f^{(i)\prime} = r \circ f^{(i)}$，其中运算符°表示两向量对应位置元素相乘（element-wise product）。以上便是 NB 特征的具体求法。Wang 在包括长文本影评 IMDB 数据集[13]、短文本影评 RT-s[14]、长文本新闻数据 20-newsgroups[2]等多个不同类型数据集上做了多个方法的对比实验，结果均证明了 NB 特征的有效性。但是文章中并未提及其中包含的道理，个人观点看来，NB 特征体现了在宏观意义上，每个特征项对于正负例划分的能力。注意到向量p、q在除去平滑因子之后，表达的内在含义是每个特征在正负例文档中

[2] http://qwone.com/~jason/20Newsgroups/

的 DF（文档频数）值。而 log-count ratio，取对数是为了规范值域，使用一范数归一为了忽略各向量绝对值大小的影响，而真正表达特征值意义的则是这两个向量之间的除法运算——对某个特征而言，这可以认为是判断这个特征在正例中的概率（注意到代表正例文档的特征向量p作为被除数在上方）。举例来说，如果某个特征项在正例文档中出现次数多，在负例中出现次数少，我们基于统计意义可以认为该特征项在正例中出现的概率大。将其相除，其较大的结果值正好表达了其概率上的含义。反过来，如果负例对正例少，那么其值较小且小于 1。如果一样多，那么相除结果为 1。注意到最后的对数运算，对 1 取对数为 0，故在正例中出现次数多的特征其 log-count ratio 值大于 0，反之小于 0，而没有区分能力的特征项其特征值为 0，这是非常好的一个性质。最后使用r更新原始布尔值表示，其实就是将原始的布尔值变为 log-count ratio 值，这样整个文档向量较原始布尔值构成的向量，有了更好的区分能力。而这其中蕴含的概率信息，则是该 NB 特征的本质，是其有更好表达能力的根本。

2.3 基于 BOW 及其衍生模型的分类器设计

上一节论述了经典文本分类方法中的文本表示部分，本节将继续沿着经典文本分类的步骤，介绍在完成文本后需要进行的分类器设计。

常用且被证明有效的分类方法包含朴素贝叶斯方法（Na ïve Bayes , NB），支持向量机(Support Vector Machine) 以及 Boosted Tree 方法。以下将分别介绍这些方法。可能根据个人理解的深浅给出详略不同的介绍。

2.3.1 朴素贝叶斯(NB)分类方法介绍

朴素贝叶斯分类方法的基本思想是利用特征项和类别的联合概率来估计给定文档属于某一类别的概率[6]424-425。

设一个文档 D 可表示为文档向量 X ， 有文档类别 c，使用贝叶斯公式表示该文档属于类别 c 的概率为$p(\mathrm{c}|\mathrm{X})$，贝叶斯分类的思想即是求使得该概率值最大的类别 c，即$arg_c \max p(c|X)$。根据贝叶斯后验概率公式展开，有

$$\mathrm{P}(\mathrm{c}\mid \mathrm{X})=\frac{\mathrm{P(c)P(X|\ c)}}{\mathrm{p(X)}}$$

其中$p(\mathrm{X})$对每一个类别 c 都一致，而$p(\mathrm{c})$可看作该类别 c 的文档数占总文档数的比例。问题的关键即是求似然概率 $p(\mathrm{X}|\mathrm{c})$的值。

而根据$p(\mathrm{X}|\mathrm{c})$求法的不同，可将朴素贝叶斯方法分为 5 类[15]。

第一种是 Multi-variate Bernoulli Na ïve Bayes 。这种方法将一篇文档的生成看

作是在由所有相异词构成的词典 D 上的伯努利实验，满足二项分布。文档使用二值化文档向量表示，即每个特征值标识该单词是否在该文档中出现。设特征项向量为$\mathrm{X} = (x_1, x_2, \ldots, x_n)^T$，特征值向量为 $\mathrm{V} = (v_1, v_2, \ldots, v_n)^T, v_i \in [0,1]$。则$p(\mathrm{X}|\mathrm{c})$表示为：

$$P(X|c) = \prod_{i=1}^{|V|} P(x_i|c)^{v_i}(1 - P(x_i \mid c))^{(1-v_i)}$$

其中$P(x_i|c) = \frac{1+N_{c,xi}}{2+N_c}$，使用拉普拉斯概率估计。$N_{c,x_i}$表示类别 C 中包含单词$x_i$的文档数。

第二种是 Multinomial Naïve Bayes，TF attributes。将文档的生成看做是在词典 D 上的有放回抽取实验。设特征项向量（即词典向量）$\mathrm{D} = (\mathrm{d}_1, \mathrm{d}_2, \ldots, \mathrm{d}_n), \mathrm{n} = |\mathrm{D}|$；文档特征值向量表示为 $\mathrm{X} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n), \mathrm{x}_i = tf_i$，$tf_i$表示词$\mathrm{d}_i$在该文档中出现频数。该文档长度$\mathrm{l} = \mathrm{len}(\mathrm{doc})$表示文档词数。则$P(X \mid c)$可表示为：

$$P(X \mid c) = P(l) * \frac{l!}{\prod_{i=1}^{n} x_i!} * \prod_{i=1}^{n} P(d_i \mid c)^{x_i}$$

其中$P(l)$表示长度为l 的概率，$\frac{l!}{\prod_{i=1}^{n} x_i!}$ 表示抽取的这些单词的可能排列组合数，$\prod_{i=1}^{n} P(d_i \mid c)^{x_i}$抽取出这些单词的概率。$P(d_i \mid c) = \frac{1+N_{di,c}}{n+N_c}$使用拉普拉斯概率估计该类别 c 下出现单词 d_i 的概率。

第三种是 Multinomial Naïve Bayes，Boolean attributes。于第二种方法唯一不同的地方即是文档特征值向量不在使用 TF 特征，而是选择该单词是否出现的布尔特征作为特征值。

第四种方法是 Multi-variate Gauss Naïve Bayes，第五种方法是 Flexible Bayes，具体未详细调研。

朴素贝叶斯分类方法在垃圾邮件过滤方面取得了巨大成功。其计算相对简单，资源消耗较低，适合在大部分客户端直接使用。同时，朴素贝叶斯也更加适用于短文本分类。

2.3.2 支持向量机(SVM)方法简介

SVM 即支持向量机(Support Vector Machine)，主要用于解决二元分类问题。如果用于多元分类，一种可能的尝试是使用 one-vs-all 方法。SVM 核心思想是希望从数据点集中找到一个决策面（Decision Boundary），并最大化该平面与数据点集的间隔，最终以此决策面划分数据。

设空间中决策平面为$y = w^T X + b$，则 SVM 的优化目标是最小化$w^T w + C\sum_i \max(0,1 - y^{(i)}(w^T x^{(i)} + b))$。[7]

SVM 方法是一种在过去时间里取得巨大成功的分类方法。其稳定和良好的效果，常被当下许多学者作为研究某一分类问题的基准线(Baseline)。同时由于其引入支持向量的机制，使得这种方法对于轻微文本类别数据偏置问题有良好的适应性。

2.3.3 NBSVM 方法简介

在[7]中，Wang 提出 NBSVM 的方法，即是 SVM with NB features 。其不仅仅使用了 NB 特征，还在 SVM 训练得到的模型之上做了细微处理。

Wang 将 SVM 分类器训练得到的权值取出，做一个插值处理来得到最终的权值向量，然后使用该权值向量做预测。具体来说，即首先插值$w' = (1 - \beta)\bar{w} + \beta w$，其中$\beta$表示插值系数，$\bar{w} = \frac{||w||_1}{|V|}$，$|V|$表示特征项的数量。预测方法为$y = sign(w^T X + b)$。文章中提及这种插值可以认为是一种正则化方式。其中的$\bar{w} = \frac{||w||_1}{|V|}$一定程度上与 L1 正则有所相似。

Wang 在文章中论述使用 NB 特征再配合插值的 NBSVM 方法在长短文本分类上均有较好表现，并用了大量的实验验证该观点。

2.3.4 Boosted Tree 方法简介

所谓 Boosted Tree，并非某一具体方法的名称，而是一系列类似方法的统称。介绍 Boosted Tree，需要首先介绍决策树(Decision Tree)及分类和回归树(Classification And Regression Tree , CART)。

决策树是一种基于树模型的层次化分类方法。对该算法用一种自然语言的方法来描述，就是每次都选择事物的一个属性并根据该属性的值将该事物进行一次划分，如果划分完成即结束，未完成则继续选择某一特征划分直到划分结束或者满足限制条件而退出。举一个简化的情感分类例子来说，假设我们的文本有 3 个词，分别是“高兴”，“难过”，“心情”，使用 BOW 模型将这三个词作为特征，并以布尔值作为特征值表示。一个可能的决策树是这样的：首先选择“高兴”这一维特征，如果特征值为 1，则直接输出结果为正例，否则进行下一轮；选择“难过”这一维特征，如果为 1，则输出结果为负例，否则进行下一轮。选择“心情”作为特征，无论为 0 或者为 1，均不能作为分类有效依据，出现这种情况，可能的情况是规定

输出为某一类别。同时为了避免这类情况的发生，决策树在构建之前一个极其重要的步骤便是特征选择，需要尽可能将区分性大的特征放在靠前的位置，如果可能，可以抛弃部分区分性不大的特征，如例子中的“心情”这维特征。上述仅是一个简单的例子，仅为了说明决策树的基本思想。实际问题中的决策树可能是非常复杂的结构，每一个节点的划分可能包含多个分支而非仅仅两个分支，树的深度也可能非常的大，为此还常常伴有预剪枝（在建树阶段评估该特征对分类的影响力，如使用熵、基尼系数等，如果其划分带来的影响低于某个阈值，则忽略该特征值的划分）与后剪枝（建树完成后根据深度的限定值去除某些分支）的情况。这里需要注意的是，通常来说，决策树处理的特征值是离散值。如果处理的特征值是连续的，且每个节点仅有两种划分结果，这种类似二叉树结构的模型被称作分类和回归树(CART)模型。可见其实 CART 和决策树差别不大，仅在处理对象和树结构上有所差异，其核心思想相同。在后续部分，将可能使用“树模型”这一词汇表示这两种树结构的统称。

使用单一的一棵树对一个复杂问题进行建模效果往往不佳。这或许是由于一颗过深的树往往对训练集太过拟合，而泛化能力较差。但能够提高泛化能力的剪枝则又太过依赖启发式规则，有时并不能带来理想的效果，并缺乏普适性。针对这一问题，一种常用的解决思路是使用 ensemble（组合）的方法。通过将多个浅层次的树组合起来作为一个整体，实现对原始问题的分类。令人惊喜的是，这种方法的效果非常的好，而且使用浅层次的树，训练代价也降低很多，泛化能力也大大提高，而且 ensemble 方法以一种数学可推导的形式给出，对于大部分问题都能适用，减少了对于启发式规则的依赖。具体来说，对树模型的 ensemble 方法也通常分为两类——将多个树并联起来，广泛应用的实现是随机森林（Random Forest , RF）；另一种便是 Gradient Boosting 方法，将树串联起来实现组合。后续将主要介绍这种方法。

所谓串联组合，就是后一棵树的预测是在前一棵树的预测之上的优化。对于前一棵树分错的数据，后一棵树加大其权值，然后对这些数据再次分类，分类结果再给下一棵树。最后，Boosting 方法将每颗树的分类结果加权合并起来，作为最终的分类结果。而 Gradient Boosting 则是指使用梯度下降的方法来构建这组串行树。

2.4 基于深度学习的文本分类方法初探

前文说到，基于 BOW 模型的文本分类方法，效果已经非常出众。正是由于这个原因，很长一段时间里，人们认为文本分类已经达到了一个上限，或转移到其他

领域，或仅关注于情感分类中涉及语义上的问题。

近年来深度学习的兴起，人们尝试将在图像处理和语音识别领域取得惊人效果的深度学习方法运用到自然语言处理领域，开辟一块新的战场。特别是在前文提及的 distributed representation[11,12]提出后，人们有了将单词转变为向量的表示，完成了从文本到类图像矩阵的转换。有了将文本表示为图像形式的能力，很自然的人们开始尝试使用原本应用于图像处理领域的深度学习模型来处理文本。Collobert 发表[16]拉开了人们跳出传统 NLP 方法，转而尝试使用深度学习方法的大幕。深度神经网络的研究热潮开始席卷自然语言处理领域。

由于时间原因，未能对深度学习领域，包括 distributed representation 有深入的了解。后文主要论述 Xiang Zhang 在[17]中使用卷积神经网络，尝试从原始字符信息中学习知识完成文本分类任务的工作，并以此作为深度学习在文本分类问题上的初探。

2.4.1 卷积神经网络简介

LeCun 等学者于 1998 年在[18]中提出了卷积神经网络(Convolutional Neural Network , CNN)这种突破传统的人工神经网络结构。传统神经网络，或者说多层感知器(Multiple Layer Perceptron , MLP) ，在输入层与输出层间，包含一个或多个隐含层(Hidden Layer)。各隐含层之间采用全连接的方式，完成特征与特征间的各种线性组合，并通过使用激活函数(Activation Function)引入非线性关系。由于全连接的关系，神经网络的参数个数非常多，不仅受限于硬件计算能力导致耗时巨大，而且当隐含层数量增多时，传统的基于后向传播的模型训练方法面临梯度稀疏的问题，模型无法得到优化。而 CNN 模型很好的解决了这个问题。

局部感受野(Local Receive Field)，权值共享，亚采样(Sub-sampling)，是 CNN 区别于传统神经网络的特性，也是能将层次做深的关键。局部感受野的思想来源于生物学，通过对猫视觉皮层的研究，人们发现一个神经元并不会处理所有的输入信息，而是仅仅关注于一个局部的区域。对应到人工神经网络上，即是各隐含层间不再是全连接的，而是后一层的一个节点仅连接前一层中某一个区域内的节点。在 CNN 中，对于一个输入层，其后一层已经不再是一个全连接的隐含层，而是由多个面构成的，每个面被称为 FeatureMap 。一个 FeatureMap 中的每个点，都是同一个过滤器(Filter)与原始输入的不同局部区域相互作用得到的结果值。一个 FeatureMap 上的所有点都是使用同一个 Filter 得到的，此为权值共享。过滤器与原始输入矩阵的所有的“不同局部区域”相互作用的过程即称作卷积(Convolution)。

上述所说的“不同局部区域”，是指不完全相同的、与过滤器等大的区域。对卷积之后的结果，往往还需要做亚采样，这个过程在 CNN 中被称作池化(Pooling)过程。所谓“池化”，就是以某个固定大小对原始图像做不重叠的亚采样——选取一个与池化大小相同的区域，取输入矩阵中在该区域内的最大值或者平均值作为该区域采样结果，然后按照该区域在原始输入中的位置关系将采样结果放入到结果矩阵中的相应位置。以上即完成了一个卷积池化过程。一般将卷积池化合并作为一个层次，可类比于 MLP 中的隐含层。CNN 网络一般包含多个这样的层次，足够深的层次使得 CNN 能够学习到足够的知识，而每个层次较 MLP 相对少了很对需要训练参数，这保证了深度学习的可能。在多个卷积池化层之后，将最后一个池化层输出的结果连接起来，作为一个浅层 MLP 的输入。到此即完成了一个 CNN 模型的构建。

2.4.2 以字符作为输入的 CNN 模型构建

深度学习的一大特点就是通过构建深层次网络，能够取代传统人工构造特征的步骤，转而依靠神经网络去学习到特征。Zhang 即以此为出发点，尝试从字符级别的原始信息入手，训练 CNN 模型完成多层次文本特征提取及分类。

由于 CNN 输入是二维矩阵，故需要将字符表示为固定长度的向量形式。Zhang 使用 1-of-m 方式选取 69 个常用字符作为字符集向量表示，并将不在该字符集内的其余字符映射为 0 向量。同时为了保证各输入矩阵等大，限制了文档的长度l_0，若长度不足l_0，则补 0 向量，超过则舍弃多余部分。

CNN 模型使用了 6 层卷积池化层，3 层 MLP（无歧义的说，即 2 层隐含层，一层输出层）。详细信息见[17]。

2.5 本章小结

本章阐述了传统文本分类方法中的 BOW 及衍生模型、基于 BOW 类模型的分类器设计，试图概述经典文本分类方法以及在经典方法上的一些改进尝试。最后，从以字符信息作为输入构建 CNN 模型完成文本分类任务这一具体方法的角度，简要介绍了 CNN 模型结构，并以此作为基于深度学习完成文本分类的初探。

第 3 章 文本分类问题中的数据偏置问题

3.1 引言

数据偏置是指在不同类别中数据量相差较大的问题，也被称作数据集不均匀、数据集不均衡等。为易懂性，后文将主要采用“数据集不均匀”这一称谓，但在某些情况下仍可能混用。

数据偏置在实际分类问题中经常出现，并往往给分类结果带来不利影响。例如在垃圾邮件过滤这一分类任务中，对于普通用户而言，垃圾邮件的数量占总邮件数量的比例一般是非常少的，如果我们将垃圾邮件作为正例，正常邮件作为负例，那么即是说这是一个正例少、负例多的数据集不均匀问题。类似的情况，还有特定基因识别，检测银行卡诈骗等。

特别地，在文本分类领域，该问题同样出现频繁。主题文本分类中，除了上述的垃圾邮件问题，新闻分类也往往容易出现不均匀问题。不同类别的新闻数量是不同的，对某个具体的新闻站点来说更加明显，因为往往某一特定站点根据用户受众会有偏好的选择、采集新闻；在某些特定时间段，如奥运会期间，各类别新闻数量也会出现非常明显的不均匀情况。在情感分类中，对于某些特定问题，往往大家的情感是有严重偏向的，这会导致数据不均匀问题。例如对一部评分很高的电影，或代表某一特定立场的群体，往往情感表达是有严重偏向的。这种偏向对于分类问题就带来了不可避免的数据不均匀问题。

在第二章中我们讨论的分类方法，在均匀数据集上一般效果很好，但是将其应用到不均匀数据集上时，往往效果出现很大的下降。在本文第四章将会通过实验验证这一论述。对于数据集不均匀情况下分类效果下滑的一种可能解释是，如果我们总体的数据集数量恰好能够让分类器在均匀数据集上学习到足够的分类规则或方法，那么在不均匀数据集上，仅有少量数据的类别并不能给分类器足够的信息，这使得分类器更加倾向于将一个问题分类为拥有大量数据的类别。这种倾向往往是对训练集里数据量的拟合，而不是对数据本身含有特征的拟合。这样在模型训练完成后，其对一个实例的分类，往往考虑到了先验的数据量对类别输出的知识，然而按照某种观点来说，总体的概率分布，对于个体而言是毫无意义的，过多倚重数量的知识，对于将一个实体分类正确在一定程度上毫无帮助。

对于不均匀数据集分类，如果按照准确率来度量，那么其结果往往非常的好，然而这仅是因为将多数少量类别的数据划分到了含有大量数据的类别。这种高准

确率对于分类问题来说是毫无意义的。这也提醒我们在面向不均匀数据集做分类时，需要采取合适的评估手段。这也是本章将要首先讨论的一个点。

由于不均匀的数据分布导致分类效果下降，我们需要尝试找到一些方法来解决或者说减轻这种影响。本章后续内容将会讨论这个问题。

3.2 数据偏置下的评价标准

在本章引言中论述了数据偏置可能导致分类器的输出结果倾向于实例数量较多的类别标签。这往往是由于将总体上各类别出现概率应用到了某个具体实例上的结果。而且如果我们依然采用准确率（统计预测正确的个数在所有待预测数量中所占的比例）来作为评价标准，其评估结果往往不能让人信服。举一个较极端的例子，假设我们有 90 个正例 10 个负例，那么只要我们的分类器给出全部为正例的输出，准确率即有$\frac{90}{90+10} * 100\% = 90\%$，然而这个足够高的结果显然并不能让我们相信这个分类器能够胜任这个分类任务。因而，面对数据集不均匀问题，我们往往不再选择准确率作为评价标准，而选择 PRF 或者 ROC 作为更有效的结果度量。

介绍 PRF 和 ROC 之前，首先介绍混淆矩阵(Confusion Matrix)。

表 3-1　用于二分类问题效果评估的混淆矩阵

	POSITIVE(实际为正例)	**NEGATIVE(实际为负例)**
POSITIVE' (分类器输出为正例)	TP	FP
NEGATIVE'(分类器输出为负例)	FN	TN

符号 TP、FP、FN、TN 的含义如表 3-1 所示，不再赘述。我们通常说的准确率，可以使用上述符号表示为$\mathrm{ACC} = \frac{\mathrm{TP+TN}}{TP+TN+FP+FN}$ 。

PRF 是在信息检索领域广泛使用评价标准，通过对某个具体类别计算结果的准确率(Precision)，召回率(Recall)，F_1 值(F_1-measure)来度量效果。需要明确的是，PRF 是类别相关的，假设我们要求取正例类别 PRF 结果，其计算方法为：

$$\mathrm{P} = \frac{\mathrm{TP}}{TP + FP}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{TP + FN}$$

$$\mathrm{F}_1 = \frac{2RP}{R + P}$$

如果要求取负例类别的 PRF，则使用与负例类别相关的数值即可。以某类别为评价主体的条件下，P 值表示了预测正确的数量在所有预测为该类别的数量中的比例，表示分类器对该类别预测的准确性评估；R 值表示预测正确的数量占实际该类别的实例数量的比例，表示一个分类器对该类别的找回能力；F_1 值则是 P 与 R 的调和平均值，是对二者效果的一个均衡。有时在特殊条件下，会遇到上述式子无意义的情况（一般是分母为 0），这时只需按照其具体含义做出该特殊情况下的定义即可，而不必理会其数学表达式的结果。对于不均匀数据集，我们一般选取少量类别作为评价主体。同样是上述 90 正例 10 负例的例子，如果分类器给出 100 个正例预测作为输出，我们采用以负例类别为主体的 PRF 评估，其 P、R 值均为 0，F 值无意义但按照实际含义，定义其为 0.

ROC 是在医药学上常用的概念，表示受试者操作特性曲线。用在分类问题评估上，通常要改变对正负例判断的临界值来得到不同的混淆矩阵，并根据这些不同值绘制一条曲线，求取曲线下的面积(Area Under Curve，AUC)来作为最终评价效果。AUC 一般越大越好。同样，ROC 也是与类别相关的。我们以正例类别为例，首先要定义 FPR(FP Rate)与 TPR(TP Rate) ：

$$\mathrm{FPR} = \frac{\mathrm{FP}}{FP + TN}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{TP + FN}$$

FPR 指在预测为正例的结果中，预测错误（实际为负例而预测为正例）的数量占实际为负例的实例总数量的比例，同理 TPR 指预测正确的数量占实际为正例的实例数量的比例，从某个意义上来说，这表示了分类器预测错误与预测正确的相对值大小。我们让分类器输出实例被预测为正例的概率，通过改变区分正例负例的概率阈值来得到一系列的(FPR , TPR)二元值，将这些值绘制在以 FPR 为横轴，TPR 为纵轴的坐标系中，即得到了 ROC 曲线。定义曲线与坐标系围成的面积表示分类效果，直观可知当预测正确的相对值高于预测错误的相对值，即$\frac{\mathrm{TPR}}{FPR} > 1$时分类器效果为正，且比值越大分类器效果越好。表现在图像走势上，就是 ROC 曲线在$y = x$上方且偏离越大越好；表现在 AUC 定量评价上，其值越大分类效果越好。如图 3-1，一个 ROC 曲线示例[3]。

由于 ROC 要求分类器的输出为概率值，这对一些分类方法（如 NBSVM）很难做到，为了后续实验有更大的可比较性，在不均匀数据集上做分类实验将采用

[3] 图片来自 http://ir.hit.edu.cn/blog/space.php?uid=13144&do=blog&id=5144

PRF 评价。

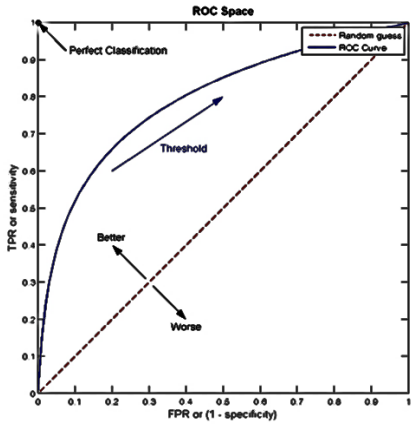

图 3-1　ROC 曲线示例

3.3 应对在数据偏置下分类效果下降的方法

在本章引言部分论述了数据偏置使分类器分类效果变差，并给出了其可能原因的直观猜测。

分类器的优化目标往往是最小化将实例错分类的代价，但同时尽可能保有一定的泛化能力，防止对训练集的过拟合。从该优化角度考虑，导致分类器在 PRF 或 ROC 评估下效果下滑的原因，包括但不限于以下两个方面：1. 分类器的泛化能力，忽略了少量数据被错误划分的损失，使其没有学习到该实例中的知识。而这些被忽略的实例有可能来自少量类别，且其相对损失更大。2.若分类器倾向优化每个被错分类的实例，那么往往带来对训练集过拟合，最终训练出的分类器因泛化能力差同样效果不好。

针对上述可能的原因，本节将论述一些应对效果下滑的方法，并尝试解释其背后可能的原理。

3.3.1 过采样与下采样

面对数据集不均匀的问题，一个最直观的想法就是尝试改变训练集，使数据集尽可能的均匀。这种思想对应的方法即是过采样(Over-sampling)与下采样(Under-sampling)。

过采样通过从实例数量较少的类别中重复有放回的抽取实例，构成一个与多

实例类别数量相匹配的数据集。下采样则相反，从多实例类别中抽取与少实例类别数量相近的实例构成新数据集。通过重新采样，从数量角度来看，已经消除了数据偏置的问题。然而实际上问题并没有得到完全解决。

Japkowicz 在[19]以及 Ling & Li 在[20]中，均提及通过有放回过采样的方法并不能显著提高分类器对少量类别的识别能力。过采样带来的提升，更多的是因为增加了少量类别的数据量而提高了其被错误分类的代价损失。然而如果分类器仅仅根据这少量被错分的数据而做出改变，那么很大可能其只是完成了对训练集上这少量数据的过拟合，而这并不能带来实质上效果提升。

下采样数据，在论文[21]中被认为是比过采样更可取的方法，能够使分类器对少量类别的数据有更大的敏感度。但是使用下采样方法，很可能导致多数量类别信息的丢失，这可能导致分类器少学习到某些特征，影响分类效果。

3.3.2 为不同类别设置不同的损失值

通过对不同类别的数据在分类器层面设置不同的损失值，可以改善数据不均匀带来的效果下滑问题。

对于 SVM 分类器，改变分类器对不同类别的权重比与过采样有相似的原理，都会使少量类别的实例被分类错误时受到更大的惩罚，因而迫使分类器调整参数以重视少量类别的实例；对于朴素贝叶斯分类器，通过改变类别的先验概率可以增加分类器对少量类别的重视。

3.3.3 基于识别的分类器

基于识别的分类器，尝试直接从少量类别中学习到信息，不使用或者使用少量其余类别的信息。

Kubat 等学者在[22]提出了名为 SHRINK 的系统，通过识别少量类别和多数类别的交叉区域来进行分类。

3.3.4 人工生成新特征（SMOTE 方法）

传统 Over-sampling 的重复采样仅仅是复制已有数据，而在[2]中，Chawla 等学者提出 SMOTE 方法，通过在几个相邻特征向量间构造新向量实现对少量类别的过采样。

SMOTE 方法的具体过程为：对某一个特征向量$\vec{x}$，利用欧式距离找到离它最近的 K 个向量，从中随机选择一个$\vec{n}$，计算两个向量的偏差$\vec{\Delta}= \vec{n} - \vec{x}$，然后在(0,1)

区间随机产生一个值α，则人工生成的新特征$\vec{s} = \vec{x} + \alpha\vec{\Delta}$ 。根据过采样比例，往往需要控制一个向量生成的新特征向量数，且尽量保证每个原始特征向量生成的人工向量数相等。

论文从决策区域选择这一角度，解释了 SMOTE 方法与传统过采样方法对决策树决策区域分割的影响。将数据空间映射到的向量空间，如果某个区域内包含的少量类别的点太少，不足以使分类器划分一个决策区域，那么这些点就将被忽略。通过复制已有数据完成过采样相当于在同一个位置有多个少量类别的点，当其数量达到一定阈值，分类器将会把该位置作为一个决策区域进行划分，然而该区域实际上只包含一个不重复点，这就导致该决策区域过于狭窄和特殊，即产生了过拟合现象。而通过 SMOTE 人工构造方法，相当于在原本稀疏的点之间增加了新的点，当这些点达到一定数量，同样使得分类器将包含这些少量类别点的区域划为新的决策区域，且这些点和原来的点构成了更大的具有更大普适性的区域，减少了过拟合的可能。

在[3]中，Akbani 通过使用 SMOTE 人工构建新的特征并使用 SVM 分类器进行分类，取得了相对 Under-sampling 更好的效果。在论文中，Akbani 提及稀疏的数据可能导致使用核函数的 SVM 产生一个过拟合的非线性分界面，然而通过 SMOTE 在各个相邻点之间人工构建新的数据点，SVM 生成的分界面将更加平滑，一定程度上减轻过拟合问题。

3.4 本章小结

本章在引言部分论述了数据偏置问题在实际问题中的常见性，并指出在数据偏置情况下大多分类器的分类效果会出现下滑。随后指出基于分类准确率的评价标准在数据不均匀条件下不再有说服力，并因此介绍了在数据偏置情况下适宜的评价方法，包括 PRF 和 ROC。最后，在本章第三小节阐述了目前常用的解决数据偏置问题的方法，包括过采样与下采样、为不同类别设置不同损失值、使用基于识别的分类器、SMOTE 方法等。

到此，文本分类方法的调研工作到此结束，后续一章是在上述调研结果指导下进行的实验验证，希望能通过真实的实验结果验证上述调研的方法，并总结出有效的针对不均匀类别的文本分类方法。

第 4 章 文本分类实验

4.1 引言

在前面两章中介绍了文本分类常用特征表示、分类器方法以及数据偏置问题和一些常用解决方案。本章将以这些内容为指导，设计完成均匀类别的文本分类实验，以实践在第二章中提及的一些文本分类方法；使用不均匀类别的数据，验证在不均匀类别下分类方法效果下滑的论述；最后以第三章介绍的一些处理方法为指导，测试应用各处理方法后分类效果是否有提升及提升程度。最后，根据总体实验结果，尝试总结出针对不均匀类别的有效分类方法。

4.2 实验数据集

英文数据集选择了在多篇论文中用作实验数据集的 IMDB 数据集[13]，该数据集属于情感分类子领域。另选择了中文数据集，在某些情况下用作额外的补充，数据源来自搜狗实验室[4]，属于主题分类子领域。

选择的数据集均为均匀数据集。一个原因在于有大量论文以此为基础的均匀数据集（IMDB 数据集）更易获取且便于不同方法下的效果比较，二则是为了论证数据集从类别均匀切换为不均匀时分类效果下滑的论述。

以下将对数据集做详细说明。

4.2.1 IMDB 数据集

在[13]中包含有一个用于情感分类的 IMDB 评论数据集，其中的评论取自 IMDB 上包含较明显极性的评论（10 分制中，评分小于等于 4 或者评分大于等于 6），同时限定每部电影最多包含 30 篇评论。数据集共包含 100,000 条评论，分为训练集、测试集各 25,000 条，此外还包含 50,000 条未标注数据。训练集与测试集均由相等比例的正负例组成。本论文中仅使用其中的训练集与测试集。

获得原始数据集后，做了转换小写、去除所有标点的预处理，得到 IMDB 数据集，见表 4-1 ：

[4] http://www.sogou.com/labs/

表 4-1　IMDB 数据集（训练集测试集 1:1，正负例 1:1）

	正例	负例
训练集	12.5k	12.5k
测试集	12.5k	12.5k

4.2.2 搜狗新闻语料

在搜狗实验室的页面中，包含有公开的 2008 年新闻数据。在老师的帮助下（官方提供的 FTP 链接一直连接不上），获得了全网新闻数据(SogouCA)和搜狐新闻数据(SogouCS)。从原始数据中根据 URL 前缀信息将新闻数据放到各自的类别，并从原始 GB18030 编码转换为 UTF-8 编码，使用 LTP[23]进行分词处理，得到处理后的数据集。由于语料庞大，包含大量新闻类别，根据问题需要从中选择了新闻数量较多的经济和娱乐两个类别作为一个数据集，又选择类别区分度较小的 IT 和科技这两个类别构成额外一个的数据集。为保证正负例数量均匀，取两个类别中新闻数量较小者作为各类别的数量，两数据集详细信息分别见表 4-2 ，4-3 。

表 4-2 搜狗经济与娱乐数据集（训练集与测试集 15:1，正负例 1:1）

	正例	负例
训练集	150k	150k
测试集	10k	10k

表 4-3 搜狗 IT 和科技数据集（训练集与测试集 3:1 ，正负例 1:1）

	正例	负例
训练集	16.5k	16.5k
测试集	5.5k	5.5k

选择 IT 和科技这两个类别构成一个数据集，是因为之前选择的经济和娱乐两个类别的新闻区分度太大，分类效果太好，不能较明显的比较各方法效果，故选择了区分度更小的 IT 和科技作为实验数据集。在后续大多数实验中，更多的使用了 IT 和科技这一数据集。将经济和娱乐类别数据集在此列出，是为了在后续实验中说明主题文本分类效果的可靠性。

4.3 均匀数据集下文本分类实验

使用第二章中调研的部分方法在上述均匀数据集上做了文本分类实验。

使用了基于 unigram 和 unigram + bigram 的布尔特征、归一化 TFIDF 特征以及 NB 特征。使用这些特征项表示维度都非常的高，故又使用了 Bag-Of-Semantic Concept 的词向量聚类方法来实现特征空间降维，以此作为与高维特征向量分类效果的对比，同时也作为将深度学习的成果与传统 BOW 模型融合的效果测试。

分类器选择上，使用了 SVM 分类器和 Boosted Tree，这两种分类器均被证明拥有良好的分类效果。因为时间及工程原因，均使用了开源实现。SVM 分类器使用 libLinear [24]，该分类器主要针对多实例、高维特征的数据，特别适合文本分类问题。Boosted Tree 使用 xgboost 工具[5]。xgboost 是在 Gradient Boosting 框架下实现的包含线性模型(generalized linear model)和梯度下降回归树(GBDT)模型的优化、可并行库，支持优化的分布式、单机并行等。其分类速度非常快。

使用 CNN 模型进行文本分类遇到了问题。模型实现上，使用 Theano[25-26]深度学习库，参照官方文档以及 CNN Sentence 代码[6]，根据[17]搭建完成 CNN 模型。在 IMDB 数据集上做了测试，数据集中平均每篇文章有 1250 个字符，由于输入长度需要满足模型的规则（保证卷积池化结果为整，不会出现边界问题），故固定长度为 1311，宽度为字符集个数 69。每个 mini-batch 包含 128 个实例，IMDB 数据集共有训练实例 25K，则共有 20 个 mini-batch。使用 5-fold 交叉验证的方法（以使用 early-stopping）。在测试运行阶段，发现该 CNN 模型耗时太长，几乎不可行——在 CPU 上跑一个轮次需要约 25 小时，在 GPU 上需要约 33 小时。耗时长久并不意外，因为在论文中也有所提及，然而在 GPU 上耗时超过 CPU，实在与常理相悖。基本可以确定代码确实在 GPU 上运行，至少启动阶段如此（使用 GPU 时会首先打印 GPU 信息）。故可能的原因一个是因为 GPU 性能，一个是代码上某些操作导致耗时严重。但由于经验缺乏，并不能找到确切的原因。同时，耗时过长导致

[5] https://github.com/dmlc/xgboost
[6] https://github.com/yoonkim/CNN_sentence

一定时间内 CNN 模型更新参数次数有限，模型调参不充分，不能达到最优效果（事实是，在 3 天时间里仅完成两轮迭代，模型完全没有拟合训练集，甚至可以认为模型保持在初始化阶段）。由于这个原因，实验最终放弃了 CNN 方法。

以下列出在 IMDB 和搜狗数据集上的实验效果，并做简要结果分析。

4.3.1 IMDB 均匀数据集实验结果

在 IMDB 原始均衡数据集上实现结果如表 4-4 所示 ：

表 4-4 IMDB 原始均匀数据集实验结果

实验方法	准确率(ACCURACY)
SVM+BOOL+UNI	86.20
SVM+BOOL+UNI_BI	**89.42**
SVM+TFIDF+UNI	88.21
SVM+TFIDF+UNI_BI	**89.74**
NBSVM+UNI	86.29
NBSVM+UNI_BI	**91.00**
BOOSTING+BOOL+UNI	86.22
BOOSTING+BOOL+UNI_BI	87.70
BOOSTING+TFIDF+UNI	86.23
BOOSTING+TFIDF+UNI_BI	87.91
BOOSTING+NB+UNI	86.22
BOOSTING+NB+UNI_BI	87.70
SVM+SEMANTIC-CONCEPT+UNI	84.81
SVM+SEMANTIC-CONCEPT+UNI_BI	85.22

注 1：由于是均衡数据集，故使用准确率作为度量；UNI 指使用 unigram，UNI_BI 指使用 unigram 和 bigram; BOOL 指布尔特征，TFIDF 指归一化的 TFIDF 特征;NBSVM 方法不仅仅使用 NB 特征，还在 SVM 训练之后做了插值，故不写为 SVM+NB 的方式;BOOTING 指第二章中提及的 Boosted Tree 方法，这里写作 BOOSTING 是因为该名称更为人熟悉。SEMANTIC-CONCEPT 即 Semantic-Concept 方法，这里为保持一致均写作大写。

注 2：参数设定上，SVM BOOL 使用 C=0.1 , NBSVM 中 C=1 , beta=0.25 , SVM TFIDF unigram 使用 C=800 ， unigram+bigram 使用 C=2100 ;BOOSTING 方法中，均设置 xgboost 弱分类器为 gbtree , max-depth=2 , num_round=2000；Semantic-Concept 设置聚类 K 值为 200

注 3：此后所有实验除非单独声明，否则均表示使用此参数设置。

从表 4-4 中可以看到，使用 NBSVM 配合 unigram 与 bigram 组合取得了最好的效果。

使用 unigram+bigram 均高于仅使用 unigram 的效果，这说明 bigram 的确增加了 BOW 模型表达语义的能力，或者说对于原始文本信息的保留能力。

而在布尔特征、归一化 TFIDF、NB 特征间，当使用 unigram+bigram 时，不论是 SVM(包括 NBSVM)或者 BOOSTING 方法效果差异其实都不大，而当仅使用 unigram 时，可以看到使用 SVM+TFIDF 效果较明显地优于其他方法约 2 个点，当然获得这样效果的前提是需要谨慎地设置 SVM 分类器参数 C 值的大小。排除这个特例，可以认为使用这三种特征其实对于分类效果影响并不大。这里需要注意，NB 特征和布尔特征在使用 BOOSTING 时，其结果是一致的，这并非偶然，而是在后续实验中均出现的结果。出现该问题的原因，是因为 BOOSTING 方法是通过每个树节点对数据进行分类的，而每一个节点都是根据一个特征的值范围来划分的。从每个特征项的角度来看，布尔特征与 NB 特征表示的特征值是成比例关系的。即布尔特征向量是 0 的位置 NB 特征向量也是 0，是 1 的位置 NB 特征向量的对应的 log-count ratio，这种性质导致 BOOSTING 方法在该两种特征向量上划分是一致的。这在后续实验中也会得以证明，在此做出解释，且后面不再赘述。

分类器比较上，使用 SVM 方法在 unigram+bigram 时效果较明显优于 BOOSTING 方法，在使用 unigram 时二者差别不大。

最后，使用 Semantic-Concept 方法降维效果还是较为理想的——将原始上万维特征降为 200 维（K=200）时，在 unigram 和 unigram+bigram 上较最好效果都仅下降 4 个点左右。这也说明了 word embedding 方法的有效性——在一定程度其的确能够表示语义信息。

4.3.2 搜狗均匀数据集实验结果

首先在经济与娱乐数据集上做了部分实验，结果如表 4-5 所示。

表 4-5 搜狗经济与娱乐均匀数据集部分实验的结果

实验方法	准确率(ACCURACY)
SVM+BOOL+UNI	99.995
SVM+BOOL+UNI_BI	99.995
NBSVM+UNI	99.955
NBSVM+UNI_BI	99.92

使用 SVM+BOOL 和 NBSVM 方法在该数据集上做了实验，发现效果非常好。这或许可以说明，在主题文本分类中，当主题类别间区分度很大时，分类器效果已经非常优秀。由于效果已经接近完全正确，故在该数据集上没有继续做实验的必要。因此选择了区分度较小的 IT 和科技类别作为中文语料上文本分类实验的数据集。

在 IT 和科技类别的实验结果如表 4-6 所示。

表 4-6 搜狗 IT 与科技均匀数据集部分实验的结果

实验方法	准确率(ACCURACY)
SVM+BOOL+UNI	97.19
SVM+BOOL+UNI_BI	98.68
SVM+TFIDF+UNI	89.25
SVM+TFIDF+UNI_BI	95.05
NBSVM+UNI	98.46
NBSVM+UNI_BI	**99.02**
BOOSTING+BOOL+UNI	98.48
BOOSTING+BOOL+UNI_BI	**98.77**
BOOSTING+TFIDF+UNI	98.47
BOOSTING+TFIDF+UNI_BI	**98.86**
BOOSTING+NB+UNI	98.48
BOOSTING+NB+UNI_BI	**98.77**
SVM+SEMANTIC-CONCEPT+UNI	96.92
SVM+SEMANTIC-CONCEPT+UNI_BI	97.15

注：参数设定上，SVM TFIDF unigram 使用 C=4096 ， unigram+bigram 使用 C=8192。其余参数与之前实验设置相同。

可以看到，即使在区分度较小的 IT 和科技类别上，分类效果依然非常出色，其中 NBSVM 配合使用 unigram 和 bigram 取得了最好效果。

依然可以看到使用 unigram+bigram 较仅使用 unigram 效果有所提升。

从特征选择这一角度可以看到，对于 BOOSTING 方法，布尔特征、NB 特征和归一化 TFIDF 特征带来的分类效果差别都不大，但归一化 TFIDF 效果最好；而使用 SVM 方法时，使用 TFIDF 特征的结果相对其他结果差距很大（unigram 条件下较最好结果差约 9 个点，unigram+bigram 上则差约 4 个点），这可以认为是实验参数设置不恰当导致的。从使用 TFIDF 特征时 SVM 参数 C 的调整来看，TFIDF 特征对于 SVM 参数调整有较强的依赖，且最优参数在不同数据集和特征项（unigram 或 unigram 与 bigram 的组合）上变化很大。这里较差的结果，很大原因是 SVM 参

数 C 不是最优的。

分类器效果比较上，SVM 与 BOOSTING 差距同样相差不大，然而从上述 TFIDF 特征来看，BOOSTING 对于这三种特征具有更好的自适应能力，不需要针对某种特征单独调整参数。

最后，在该数据集上，Semantic-Concept 方法依然表现出很高的性价比。

4.3.3 均匀数据集下实验结果总结

在上述两个数据集上，一致的结论是使用 unigram 和 bigram 的组合的确均带来了分类效果的提升。然而可以看到在 IMDB 上带来的提升较在搜狗 IT 和科技类别上的提升更大。一种可能的解释是 bigram 在情感分类上带来的帮助较主题分类上更加显著，这在论文[7]中有所提及。当然，这种解释并非有让人完全信服的能力，毕竟在搜狗数据集上使用 unigram 已经达到了非常好的效果，想要在如此好的效果上带来提升本身是更加有难度的。同时，上述实验结果并未展示特征维度的变化，实际上，在 IMDB 数据集上，使用 unigram 特征向量维度越 6 万，而使用 unigram 与 bigram 的组合，则达到了近 100 万，这带来的时间和空间开销是必须被考虑的。

布尔特征、归一化 TFIDF 特征及 NB 特征，在 BOW 模型下都是有效的，且彼此带来的效果差异并不显著。这或许是由于在未做特征选择的条件下，标识一个文档中出现的足够多的词已经能够表明文档的类别信息，不必额外的信息。故布尔特征取得的效果已足够的好。 当然如果关注明确的效果好坏，那么布尔特征与归一化 TFIDF 和 NB 特征相比仍然是有差距的，而归一化 TFIDF 特征与 NB 特征之间在不同分类器下表现不一，且归一化 TFIDF 特征在 SVM 分类器下对参数较为敏感，需要谨慎设置参数。

分类器上，在均衡数据集下 SVM 与 BOOSTING 效果均非常优异（当然，这里缺乏一个对比，但从实际效果来看可以这么说）。但是从归一化 TFIDF 特征可以看出，SVM 对于参数设置是更加依赖的，即不同数据集需要设置不同的参数，而 BOOSTING 则具有神奇的自适应能力。

最后，Semantic-Concept 方法具有很高的性价比——效果下降不大，但所需时间与空间开销大幅降低。不过，Semantic-Concept 在训练模型时需要更多的预处理，包括词向量生成，Ngram 词向量表示，词向量聚类，特征值计算等。然而只要完成模型训练，后续测试或预测均与其他方法相差不大。

4. 4 不均匀数据集下文本分类实验

本节先论述不均匀数据集的构建方法，随后给出直接在该不均匀数据集上使用上述文本分类方法的实验结果。

实验结果使用少量类别的 F_1 值作为度量，同时，为了比较由均匀数据集切换到不均匀数据集时的效果变化，列出均匀数据集下相应类别的 F_1 值作为对比。

4.4.1 不均匀数据集

为了构建不均匀数据集，分别将 IMDB 和搜狗 IT 和科技数据集中的正例类别随机取样（不放回），构建了正负例比例 10:1 的不均匀数据集。表 4-7,4-8 分别列出了不均衡处理后的两个新数据集信息。

表 4-7 IMDB 不均匀数据集（训练集测试集 1:1，正负例 10:1）

	正例	负例
训练集	12.5k	1.25k
测试集	12.5k	1.25k

表 4-8　IT 和科技不均匀数据集（训练集与测试集 3:1 ，正负例 10:1）

	正例	负例
训练集	16.5k	1.65k
测试集	5.5k	0.55k

上述数据集将作为后续不均匀实验的数据集。

4.4.2 直接在不均匀数据集上进行文本分类实验

对上述数据集不采取任何措施，直接应用上述文本分类方法，以少量类别的 F_1 值作为度量，同时列出均衡数据集下该类别的 F_1 值作为对比，得到实验结果如下。

4.4.2.1 IMDB 不均匀数据集上分类效果

在 IMDB 均衡数据集和正负例 10:1 的不均衡数据集上实验，以负例类别（不均衡数据集中的少量数据类别）的 F_1 值作为度量，实验结果如表 4-9 所示。

表 4-9 在 IMDB 均衡和正负例 10:1 的不均匀数据集上以负例类别 F_1 值为度量的实验结果.

实验方法	F_1 值(均衡)	F_1 值(不均衡)	Δ
SVM+BOOL+UNI	86.23	**61.32**	-24.91
SVM+BOOL+UNI_BI	89.38	**60.61**	-28.78
SVM+TFIDF+UNI	88.26	47.29	-40.97
SVM+TFIDF+UNI_BI	**89.67**	16.97	-72.70
NBSVM+UNI	86.70	51.22	-35.48
NBSVM+UNI_BI	**91.04**	24.77	-66.27
BOOSTING+BOOL+UNI	86.23	**59.92**	-26.31
BOOSTING+BOOL+UNI_BI	87.62	59.79	-27.83
BOOSTING+TFIDF+UNI	86.14	56.92	-29.22
BOOSTING+TFIDF+UNI_BI	87.81	58.38	-29.43
BOOSTING+NB+UNI	86.23	**59.92**	-26.31
BOOSTING+NB+UNI_BI	87.62	59.79	-27.83
SVM+SEMANTIC-CONCEPT+UNI	84.67	53.49	-31.18
SVM+SEMANTIC-CONCEPT+UNI_BI	84.96	11.14	-73.82

4.4.2.2 IT 和科技不均匀数据集上分类效果

在 IT 和科技类别均衡数据集和正负例 10:1 的不均衡数据集上做了相同实验，结果如表 4-10 所示。

表 4-10 在 IT 和科技均衡及正负例 10:1 的不均匀数据集上的实验结果.

实验方法	F_1 值(均衡)	F_1 值(不均衡)	Δ
SVM+BOOL+UNI	97.21	83.75	-13.46
SVM+BOOL+UNI_BI	98.69	**92.91**	-5.78
SVM+TFIDF+UNI	90.12	20.30	-69.82
SVM+TFIDF+UNI_BI	95.15	27.85	-67.30
NBSVM+UNI	98.47	82.13	-16.14
NBSVM+UNI_BI	**99.02**	90.26	-8.76
BOOSTING+BOOL+UNI	98.49	91.45	-7.04
BOOSTING+BOOL+UNI_BI	**98.78**	**93.24**	-5.54
BOOSTING+TFIDF+UNI	98.48	89.81	-8.67
BOOSTING+TFIDF+UNI_BI	**98.87**	**93.74**	-5.13
BOOSTING+NB+UNI	98.49	91.45	-7.04
BOOSTING+NB+UNI_BI	**98.78**	**93.24**	-5.54
SVM+SEMANTIC-CONCEPT+UNI	96.89	79.55	-17.34
SVM+SEMANTIC-CONCEPT+UNI_BI	97.11	84.92	-12.19

4.4.3 直接在不均匀数据集上应用分类方法的实验结果分析

直观的看出，在 IMDB 不均衡数据集上分类效果出现了大幅下滑。F_1 值下降的背后，召回率降低是其主要原因。在 IT 和科技不均衡数据集上，分类效果尽管同样出现下滑，但下滑幅度不大。

在 IMDB 不均衡数据集上，加入 bigram 后几乎所有方法的分类效果都反而出现了下降，而在 IT 和科技类别上依然带来了效果提升。背后的原因不是很清晰，或许是由于情感分类和主题分类面临的对象是不同的，新闻主题分类上，即使负例类别数量较少，但仍然覆盖了绝大部分负例的信息，而情感分类则需要更多数据来防止分类器仅对训练集中的负例数据过拟合。表达情感的文本是更多变的，而新闻主题的词语相对固定。

在布尔特征、归一化 TFIDF 特征和 NB 特征中，原来表现良好的 NBSVM 方法在不均衡数据集上效果大幅下降，归一化 TFIDF 在 SVM 分类器下表现非常差，而布尔特征在 SVM 分类器下表现最好，说明布尔特征对不均衡数据集更有适应性，而 NBSVM、归一化 TFIDF 特征在 SVM 方法下对数据集是敏感的，直接应用在不

均衡数据集上表现不佳。这三种特征在 BOOSTING 下均表现稳定且良好。

分类器对比上，SVM 在布尔特征上表现稳定，但在其他两个特征下表现非常差。而 BOOSTING 方法表现更加稳定。特别地，我们对比 unigram+布尔特征在 IMDB 数据集和搜狗 IT 和科技数据集上 SVM 方法与 BOOSTING 方法的效果。在 IMDB 上 SVM 方法较 BOOSTING 好 0.4 个点，但在搜狗数据上差 7.7 个点。注意到这两个实验 SVM 方法使用相同的参数，这或许说明在跨数据集时，如果不相应的调整 SVM 参数，那么 SVM 分类效果并不会很优异。反过来这说明了 BOOSTING 方法的自适应性很强。

Semantic-Concept 方法表现很差。一方面可能是受 SVM 分类器影响，一方面也可能是在不均匀数据集下使用词聚类会把少量类别的词语信息淹没。

4.5 不均匀数据集下使用过采样或下采样处理后的文本分类实验

通过上一节实验，我们明确了在数据不均匀条件下，原来表现优秀的分类方法效果出现下滑。我们首先尝试使用过采样或下采样的方法来构建均匀的数据集，然后再应用各分类方法并记录其试验效果。

过采样与下采样都是通过随机有放回抽取原始数据来构建的。

由于时间原因，同时考虑到在搜狗 IT 和科技不均衡数据集上部分分类方法的分类效果已足够好，故实验仅在 IMDB 不均衡数据集上测试，后续实验同样如此，且后文不再赘述。

4.5.1 过采样少量类别数据的文本分类实验

在表 4-7 所示的 IMDB 不均衡数据集上对负例类别过采样得到 1:1 均衡数据集，如表 4-11 所示。

表 4-11 过采样负例数据得到训练集 1:1 均衡的 IMDB 数据集.

	正例	负例
训练集	12.5k	12.5k
测试集	12.5k	1.25k

注意，仅过采样训练集，测试集保持不变。

在此过采样数据集上的实验结果如表 4-12 所示。列出在不经处理的不均衡数据集上的实验结果作为对照。

表 4-12 在不均匀与过采样处理的 IMDB 数据集上的实验效果对比.

实验方法	F_1值(原始不均衡数据集)	F_1值(过采样)	Δ
SVM+BOOL+UNI	**61.32**	61.39	+0.07
SVM+BOOL+UNI_BI	**60.61**	60.68	+0.07
SVM+TFIDF+UNI	47.29	56.05	+8.76
SVM+TFIDF+UNI_BI	16.97	**67.47**	**+50.50**
NBSVM+UNI	51.22	**62.84**	+11.62
NBSVM+UNI_BI	24.77	47.43	**+22.66**
BOOSTING+BOOL+UNI	**59.92**	61.59	+1.67
BOOSTING+BOOL+UNI_BI	59.79	**63.86**	+4.07
BOOSTING+TFIDF+UNI	56.92	58.31	+1.39
BOOSTING+TFIDF+UNI_BI	58.38	62.04	+3.66
BOOSTING+NB+UNI	**59.92**	61.59	+1.67
BOOSTING+NB+UNI_BI	59.79	**63.86**	+4.07
SVM+SEMANTIC-CONCEPT+UNI	53.49	46.09	-7.40
SVM+SEMANTIC-CONCEPT+UNI_BI	11.14	30.63	**+19.49**

4.5.2 下采样多数类别数据的文本分类实验

在表 4-7 所示的 IMDB 不均衡数据集上对正例类别数据下采样得到 1:1 均衡数据集，如表 4-13 所示。

表 4-13 下采样正例数据得到训练集 1:1 均衡的 IMDB 数据集.

	正例	负例
训练集	1.25k	1.25k
测试集	12.5k	1.25k

在此数据集上的实验结果如表 4-14 所示。同样列出在不经处理的不均衡数据集上

的实验结果作为对照。

表 4-14 在不均匀与下采样处理的 IMDB 数据集上的实验效果对比.

实验方法	F_1值(原始不均衡数据集)	F_1值(下采样)	Δ
SVM+BOOL+UNI	**61.32**	45.70	-15.62
SVM+BOOL+UNI_BI	**60.61**	**49.23**	-11.38
SVM+TFIDF+UNI	47.29	**52.92**	+5.63
SVM+TFIDF+UNI_BI	16.97	48.10	**+31.13**
NBSVM+UNI	51.22	47.85	-3.37
NBSVM+UNI_BI	24.77	**53.87**	**+29.10**
BOOSTING+BOOL+UNI	**59.92**	43.02	-16.90
BOOSTING+BOOL+UNI_BI	59.79	43.47	-16.32
BOOSTING+TFIDF+UNI	56.92	45.17	-14.62
BOOSTING+TFIDF+UNI_BI	58.38	45.00	-13.38
BOOSTING+NB+UNI	**59.92**	43.02	-16.90
BOOSTING+NB+UNI_BI	59.79	43.47	-16.32
SVM+SEMANTIC-CONCEPT+UNI	53.49	42.33	-11.46
SVM+SEMANTIC-CONCEPT+UNI_BI	11.14	48.94	**+37.80**

4.5.3 过采样和下采样处理后文本分类实验结果分析

从效果提升上来看，使用过采样对 F_1 基本都有提升，然而使用下采样却使得原来表现较好的方法效果变差，虽然有部分方法提升较大，但实际值仍然很低，故效果不好。在该数据集上来看，使用过采样方法是更好的选择。

在过采样条件下，关注 unigram 下各特征及分类器表现。NBSVM 方法效果相对大幅提升，有最好的实验效果；使用布尔特征下 BOOSTING 方法提升较 SVM 大且最终结果超过 SVM 方法；TFIDF 效果相对较差。

当过采样配合 unigram 与 bigram 的 TFIDF 特征时，SVM 方法取得了最好的效果。这说明 TFIDF 在均衡数量时，如果拥有足够的信息（加入 bigram），其表达能力的优势就能够在 SVM 分类器下体现出来。然而从不均匀数据集到过采样后均匀数据集的巨大提升，进一步说明使用归一化 TFIDF 特征的 SVM 方法对数据集是非常敏感的，且在均衡数据集下表现不错，而在不均衡时表现很差。

4.6 不均匀、过采样和下采样数据下调整 SVM 参数的实验

SVM 分类器可以通过调节 C 值和各类别的权重 W 为各个类别设置不同的 Cost，同时设置偏移值 bias 能够决定决策平面是否过原点。对于 NBSVM 方法，还可以调节插值系数 beta 来改变预测模型。

对使用布尔特征，归一化 TFIDF 特征的 SVM 方法及 NBSVM 进行参数调整，主要通过枚举及二分法手工调参，由于工作量较大且难以保证结果为全局最优，故最终调参结果是相对更优的结果。

调参通过在训练集上使用多轮交叉验证的方式进行，但这可能导致调整出的参数对训练集过拟合。

分别在原始不均衡数据集（见表 4-7），过采样数据集（见表 4-11）和下采样数据集（见表 4-13）上进行调参。结果分别见表 4-15，4-16，4-17，列出未调参时的结果作为对比，列出最终使用的参数作为记录。

表 4-15 在不均匀数据集上调整参数的实验结果.

实验方法	F_1 值(原始不均衡数据集)	F_1 值(调参)	Δ	参数值
SVM+BOOL+UNI	**61.32**	63.61	+2.29	c=0.005;bias=1 w_p=1;w_n=1
SVM+BOOL+UNI_BI	**60.61**	**66.75**	+6.14	c=00075;bias=-1 w_p=1;w_n=15
SVM+TFIDF+UNI	47.29	**65.67**	**+18.38**	c=170; bias=-1 w_p=1;w_n=10
SVM+TFIDF+UNI_BI	16.97	**68.82**	**+51.85**	c=500;bias=-1 w_p=1;w_n=12.5
NBSVM+UNI	**51.22**	63.15	+11.93	c=0.001; bias=-1 w_p=1;w_n=10 beta=0.125
NBSVM+UNI_BI	24.77	55.35	**+30.58**	c=0.0001; bias=-1 w_p=1;w_n=15 beta=0.125

注：参数栏中，w_p 指正例类别的权重，w_n 指负例类别的权重，下同。对 libLinear，每个类别的 Cost 为 W*C，如正例的权重就为 w_p*c .

表 4-16　在经过过采样处理的数据集上调整参数的实验结果.

实验方法	F_1 值(过采样)	F_1 值(调参)	Δ	参数值
SVM+BOOL+UNI	61.39	**60.89**	**-0.50**	c=0.01;bias=-1 w_p=14;w_n=1
SVM+BOOL+UNI_BI	60.68	**57.68**	-3.00	c=001;bias=1 w_p=15;w_n=1
SVM+TFIDF+UNI	56.05	56.02	**-0.03**	c=950; bias=-1 w_p=1;w_n=10
SVM+TFIDF+UNI_BI	**67.47**	**68.89**	**+1.42**	c=950;bias=-1 w_p=1;w_n=1
NBSVM+UNI	**62.84**	45.19	-17.65	c=10; bias=1 w_p=1;w_n=1 beta=0.5
NBSVM+UNI_BI	47.43	17.52	-29.91	c=0.1; bias=1 w_p=5;w_n=1 beta=0.5

表 4-17　在经过下采样处理的数据集上调整参数的实验结果.

实验方法	F_1 值(下采样)	F_1 值(调参)	Δ	参数值
SVM+BOOL+UNI	45.70	40.11	-5.59	c=0.01;bias=1 w_p=1;w_n=5
SVM+BOOL+UNI_BI	**49.23**	45.92	-3.31	c=01;bias=-1 w_p=1;w_n=10
SVM+TFIDF+UNI	**52.92**	**51.74**	**-1.18**	c=450; bias=-1 w_p=1;w_n=1
SVM+TFIDF+UNI_BI	48.10	41.86	-6.24	c=550;bias=-1 w_p=1;w_n=1
NBSVM+UNI	47.85	**47.47**	**-0.38**	c=1; bias=-1 w_p=1;w_n=5 beta=0.25
NBSVM+UNI_BI	**53.87**	**53.78**	**-0.09**	c=0.1; bias=-1 w_p=1;w_n=1 beta=0.5

4.6.1 参数调整的实验结果分析

在原始不均衡数据集上调整 SVM 分类器参数，均取得了结果的提升。其中使用归一化 TFIDF 特征的分类方法效果提升最大，且在使用 unigram 和 bigram 组合时取得所有方法中最好效果，仅使用 unigram 时效果也很好。

在过采样处理的数据集上调整参数，除 TFIDF+UNI_BI 这一项，其余效果均下滑。然而其在训练集上的交叉验证结果是非常高的，如下滑最大的 NBSVM+UNI_BI 项，其交叉验证结果 F_1 值达到了 99.99%，而最终在测试集上使用该模型准确率达到 100%，然而召回率却仅有 9.6%，导致 F_1 值很低。这说明该方法对训练集出现了比较严重的过拟合现象。但采用 TFIDF 特征的方法表现有所不同，其在训练集上同样存在过拟合现象，但由于其在测试集上的准确率与召回率相差不大，故 F_1 值反而较高，如最优效果的 TFIDF+UNI_BI 一项，其在测试集上准确率与召回率分别为 71.85%、66.16%，较为均衡。

而对于下采样处理的数据，调整参数并在训练集上交叉验证寻找最优参数，同样导致了过拟合训练集的结果，而且往往在训练集上的交叉验证结果很好，其在测试集上表现就很差。

综合三个数据集来看，使用过采样处理+TFIDF 特征+unigram 与 bigram 组合 SVM 方法在调参之后取得了最好结果，不过相较于直接在原始不均匀数据集上使用 TFIDF 特征及 unigram 和 bigram 方法调参，提升 0.07 个点，可以忽略不计。这说明仅需参数调整就几乎能让使用 TFIDF 特征+unigram 和 bigram 的 SVM 方法取得比其他方法更好的效果。

4.7 不均匀数据集下使用 SMOTE 方法的文本分类实验

SMOTE 方法不是在数据空间进行采样，而是在特征空间上从两个相近向量间构建一个新的特征向量。某种程度上，这也可认为属于过采样方法。

首先将 IMDB 不均匀数据在各特征表示下转换为向量形式，然后再以向量的欧式距离度量，选择离每个负例最近的 18 个负例，从这 18 个负例中随机有放回抽样 9 个作为目标点，然后分别随机一个小数在该负例与各目标点间构建一个中间点。最后即对原始负例数据生成了 9 倍的人工数据。将其加入到训练集中，构建了对分类器而言的均衡数据集。

在该数据集上的分类效果见表 4-18.

表 4-18 使用 SMOTE 方法的实验结果.

实验方法	F_1值(原始不均衡数据集)	F_1 值(SMOTE 方法)	Δ
SVM+BOOL+UNI	**61.32**	**61.23**	-0.09
SVM+BOOL+UNI_BI	**60.61**	**60.77**	+0.16
SVM+TFIDF+UNI	47.29	**63.58**	**+16.29**
SVM+TFIDF+UNI_BI	16.97	52.25	**+35.28**
NBSVM+UNI	51.22	52.15	+0.93
NBSVM+UNI_BI	24.77	24.77	+0.00
BOOSTING+BOOL+UNI	**59.92**	59.21	-0.71
BOOSTING+BOOL+UNI_BI	59.79	60.18	+0.39
BOOSTING+TFIDF+UNI	56.92	56.61	-0.31
BOOSTING+TFIDF+UNI_BI	58.38	57.51	-0.87
BOOSTING+NB+UNI	**59.92**	59.73	-0.19
BOOSTING+NB+UNI_BI	59.79	59.95	+0.16
SVM+SEMANTIC-CONCEPT+UNI	53.49	55.17	**+1.68**
SVM+SEMANTIC-CONCEPT+UNI_BI	11.14	11.99	+0.85

4.7.1 使用 SMOTE 方法的实验结果分析

使用 SMOTE 方法的实验结果并没有直接过采样数据效果好。通过 SMOTE 方法构建均匀数据集，在使用 SVM 方法的归一化 TFIDF 特征表示下有较大提升，但对其他方法影响较小，且影响较为随机，并不稳定。

在 BOOSTING 方法下，由之前的实验及相应的理论分析知，对于同样数据集，使用布尔特征和 NB 特征其结果应该是相同的，在这里却有了小幅的差异，这是因为人工构造点的加入打破了原始特征表示所具有的特性。

4.8 本章小结

本章以第二章和第三章内容为指导，在时间条件允许的情况下使用了尽可能多的实验方法来验证调研内容。

实验结合多种变化条件，包含 unigram 及 unigram 和 bigram 的组合，布尔特征、归一化 TFIDF 特征、NB 特征，SVM 与 BOOSTING 分类方法，再加上使用 Semantic-Concept 的 SVM 分类方法，总计共包含 14 种分类方法。将这些方法应用

在均衡 IMDB 数据集，搜狗 IT 和科技数据集，以及在不均衡 IMDB 数据集上实验过采样、下采样及 SMOTE 方法，对使用 SVM 分类器的方法调参，最终得到了相对丰富的实验结果。

实验结果首先表明在不均匀数据集下，原来表现良好的分类器效果出现下滑。而使用布尔特征的分类效果在不均匀数据下效果更加稳定优秀。实验接着尝试过采样、下采样以及对 SVM 分类器调参，发现对于 SVM 方法，仅需恰当调整分类器参数就能取得较好效果，而对于 BOOSTING 方法，过采样能确保在原有结果上有小幅提升。最后尝试使用 SMOTE 方法在空间维度构建新数据点，发现效果并不明显。

实验结果表明，使用 BOOSTING 方法结合布尔特征，加上 unigram 和过采样就能取得稳定而不错的效果，且具有自适应性。而 SVM 方法则适合针对数据集调整参数，使用归一化 TFIDF 特征和 unigram+bigram 表示来获得更好的效果。

结　论

通过对文本分类方法及文本分类中数据偏置问题的调研，总结得到了一些面向不均匀类别的文本分类方法，并在时间允许的情况下对部分方法做了较为充分的实验，以下将分点论述实验结果、研究内容的创新性成果、进一步在本研究方向进行研究的工作展望与设想。

（1）**实验结果**　　实验将特征项、特征值、分类器作为影响分类效果的因子，设计完成了特征项包含 unigram 与 unigram+bigram，特征值包含布尔特征、归一化 TFIDF 特征和 NB 特征，分类器包含 SVM 和 BOOSTING 的所有可能的组合实验。此外增加了基于 word embedding 的 Bag-Of-SemanticConcept 实验和将 CNN 用于文本分类的实验。实验数据集选择了英文的 IMDB 数据集和中文的搜狗 IT 和科技数据集，分别属于文本分类问题中的情感分类和主题分类子问题。在不均匀类别的数据处理上，使用了过采样或下采样技术，并对 SVM 方法尝试调整分类器参数，最后再测试了使用 SMOTE 方法人工构建新特征向量的效果。除 CNN 方法由于耗时太长未能完成外，其余方法均在该实验设置下得出了相应结果。

实验结果表明，面对不均匀类别的数据集，加入 bigram 特征项并不能保证实验效果的提升，但结合其他不均匀数据集的处理方法往往能在分类效果上带来一定增益。在研究问题中可以将其作为一个提升分类效果的尝试，但在实际可用系统中加入 bigram 仍需要根据系统目标、条件来考虑。特征项选择上，布尔特征具有非常好的适应性。在均匀与不均匀数据集、SVM 或 BOOSTING 分类器上使用布尔特征均获得了稳定而出色表现，适合作为研究问题中 baseline 方法的特征选择，并适合在实际问题中快速构建鲁棒性良好的分类系统。对于 NB 特征，当使用 BOOSTING 分类器时，其与布尔特征等价，而当使用以 SVM 为基础的 NBSVM 方法时，NB 特征（更准确的说，是 NBSVM 分类方法）并不适合不均匀类别的数据集。归一化 TFIDF 特征无疑有更好的表达能力，但配合使用 SVM 时其太过敏感，需要足够耐心地进行参数调整和（或）训练集预处理。适合在研究问题中测试分类器参数及不均衡数据对分类的影响，在实际问题中适合作为提高分类效果的尝试。分类器选择上，BOOSTING 是鲁棒性、适应性更好的选择，而 SVM 在良好的数据集和恰当的参数设置下效果更出色。数据偏置的确对分类效果带来了很大的影响，而面对不均匀类别的数据，在过采样、下采样、参数调整、SMOTE 等方法中，基于随机有放回抽取的过采样与参数调整是更好的处理方式。最后，Bag-Of-SemanticConcept 方法作为使用 word embedding 进行特征向量降维，并将深度学习

结果与传统模型结合的崭新尝试，取得了较为理想的效果，这有效证明了 word embedding 在语义表达上的能力，作为研究问题也许值得继续探索，在实际问题上也有一定性价比。

以上述实验结果为基础，我们提出两种设计方案。设计 1 使用 unigram+布尔特征+BOOSTING+过采样方法，可快速构建稳定性高自适应强效果不差的分类系统；而设计 2 使用归一化 unigram 和 bigram 的 TFIDF 特征，选择 SVM 分类器，需要针对数据集恰当地调整 SVM 参数，这可能提供更好的分类效果，且不必对数据集做任何处理。

实现上，BOOSTING 方法可以选择 xgboost 开源实现，SVM 则建议选择 libLinear 分类器。其余特征抽取、训练数据预处理较为简单，且代码已公布在互联网上[7]。

（2）**研究内容的创新性成果**　论文在创新性上未有亮点，全文主要着重对现有方法的调研以及实验，并给出针对不均匀类别的分类方法选择。

（3）**在本研究方向上的工作展望与设想**　工作展望上，主要关注 Bag-Of-SemanticConcept 方法是否有提高的可能，因为在特征值、分类器选择上该方法还有更多的改进可能。而对未能得出实验结果的 CNN 方法，则留下了想象的空间。CNN 方法在图像和语音处理上取得了令人瞩目的成果，而目前广大的学者正尝试将其应用到自然语言处理上来。这或许又会是一场领域的革新。

[7] https://github.com/memeda/GriduationDesignCodeForTextClassification

参考文献

[1] Mani I, Zhang I. kNN approach to unbalanced data distributions: a case study involving information extraction[C]//Proceedings of Workshop on Learning from Imbalanced Datasets. 2003.

[2] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of artificial intelligence research, 2002: 321-357.

[3] Akbani R, Kwek S, Japkowicz N. Applying support vector machines to imbalanced datasets[M]//Machine Learning: ECML 2004. Springer Berlin Heidelberg, 2004: 39-50.

[4] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[5] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks[C]//Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013: 6645-6649.

[6] 宗成庆.统计自然语言处理[M].第二版.北京：清华大学出版社，2013:416-433

[7] Wang S, Manning C D. Baselines and bigrams: Simple, good sentiment and topic classification[C]//Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2. Association for Computational Linguistics, 2012: 90-94.

[8] Deerwester S C, Dumais S T, Landauer T K, et al. Indexing by latent semantic analysis[J]. JAsIs, 1990, 41(6): 391-407.

[9] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. the Journal of machine Learning research, 2003, 3: 993-1022.

[10] Lebret R, Collobert R. N-gram-Based Low-Dimensional Representation for Document Classification[J]. ICLR Workshop , 2015

[11] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[12] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in Neural Information Processing Systems. 2013: 3111-3119.

[13] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In Proceedings of ACL.

[14] Pang B, Lee L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts[C]//Proceedings of the 42nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2004: 271.

[15] Metsis V , Androutsopoulos I , and Paliouras G . Spam Filter with Na ïve Bayes – Which Na ïve Bayes ?[C]. CEAS 2006 – Third Conference on Email and Anti-Spam , 2006

[16] Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch[J]. The Journal of Machine Learning Research, 2011, 12: 2493-2537.

[17] Zhang X, LeCun Y. Text Understanding from Scratch[J]. arXiv preprint arXiv:1502.01710, 2015.

[18] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[19] Japkowicz N. The class imbalance problem: Significance and strategies[C]//Proc. of the Int'l Conf. on Artificial Intelligence. 2000.

[20] Ling C X, Li C. Data Mining for Direct Marketing: Problems and Solutions[C]//KDD. 1998, 98: 73-79.

[21] Domingos P. Metacost: A general method for making classifiers cost-sensitive[C]//Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1999: 155-164.

[22] Kubat M, Holte R C, Matwin S. Machine learning for the detection of oil spills in satellite radar images[J]. Machine learning, 1998, 30(2-3): 195-215.

[23] Che W, Li Z, Liu T. Ltp: A chinese language technology platform[C]//Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations. Association for Computational Linguistics, 2010: 13-16.

[24] Fan R E, Chang K W, Hsieh C J, et al. LIBLINEAR: A library for large linear classification[J]. The Journal of Machine Learning Research, 2008, 9: 1871-1874.

[25] Bastien F, Lamblin P, Pascanu R, et al. Theano: new features and speed improvements[J]. arXiv preprint arXiv:1211.5590, 2012.

[26] Bergstra J, Breuleux O, Bastien F, et al. Theano: a CPU and GPU math expression compiler[C]//Proceedings of the Python for scientific computing conference (SciPy). 2010, 4: 3.

哈尔滨工业大学本科毕业设计（论文）原创性声明

本人郑重声明：在哈尔滨工业大学攻读学士学位期间，所提交的毕业设计（论文）《面向不均匀类别的文本分类系统设计与实现》，是本人在导师指导下独立进行研究工作所取得的成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明，其它未注明部分不包含他人已发表或撰写过的研究成果，不存在购买、由他人代写、剽窃和伪造数据等作假行为。

本人愿为此声明承担法律责任。

作者签名：　　日期：　　年　　月　　日

致　谢

衷心感谢导师***副教授对本人的精心指导。他高屋建瓴，对课题给予热切的指导和关注，保证了毕业设计能够持续稳定地进行并最终顺利完成。他的言传身教将使我终生受益。

感谢实验室**博士。他与车万翔副教授共同给出了详细的毕业设计实施方案。他远在国外，在百忙之中抽出时间对本人给予详细指导，再次表示衷心感谢。

感谢实验室**师兄、**师兄、**师兄在具体细节问题上答疑解惑。感谢**师兄对本人的指导和精神上的鼓励。感谢实验室 LA 组师兄师姐及同级同学的帮助。感谢实验室全体老师和同学！

最后，向在文本分类、机器学习领域刻苦专研并共享知识的研究者致敬！

附录 1

Baselines and Bigrams: Simple , Good Sentiment and Topic Classification

Sida Wang and Christopher D.Manning

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2. Association for Computational Linguistics, 2012: 90-94.

基准线和二元模型：简单，优秀的情感和主题分类

摘要

朴素贝叶斯(NB)和支持向量机(SVM)的变体通常被作为文本分类的基准方法但是他们的表现往往因为模型变体、使用的特征和具体任务的数据不同而差别很大。在本篇论文中，我们将要展示：1. 在情感分析任务上使用二元模型总是能够带来稳定的增益。2. 对于短文本，朴素贝叶斯方法的效果的确要优于支持向量机（反过来，对于长文本，支持向量机的分类效果要优于朴素贝叶斯方法）。 3. 使用基于朴素贝叶斯的 log-cout ratio 值作为特征，这种简单而新颖的支持向量机变体在跨任务和数据集上效果均非常优秀。基于实验数据观察，我们确认这种简单的朴素贝叶斯和支持向量机的变体在情感分析数据集上的效果要超过大多数公开的结果，有时甚至达到了业界领先的表现水平。

简介

朴素贝叶斯(NB)和支持向量机模型(SVM)在文本分类和情感分析研究中通常作为其他方法的基准线。但是，它们的表示强烈的依赖于变体类型、使用的特征以及何种数据集。我们注意到研究者们并没有足够重视这些模型的选择问题。事实上，我们展示了一个好的变体的分类表现往往能够在很多数据集上打败最近公开的领先结果。我们试图对何种方法、何种变体、何种特征在何种环境下表现更好作出概括性的总结。

首先，我们要强调情感分析和主题文本分类存在有巨大的不同。二元模型在情感分类问题上的作用被忽视，可能就是因为人们混淆了它在主题文本分类上的鸡肋表现。接着我们要区分短文本的情感分类任务和长评论的文本分类。我们将要展示，在短文本上，朴素贝叶斯方法是要优于支持向量机方法的。另一方面，我们要

展示词袋模型在短文本中仍然有非常好的效果，结合使用朴素贝叶斯模型其分类效果往往能够超过那些最近才提出的结构精巧而复杂的模型。而且，通过将生成模型和判别模型相结合，我们提出了一个简单的模型变体。该模型变体是以融合了朴素贝叶斯思想的 log-count ratio 为特征值而建立起来的 SVM 模型。该模型在所有论文中呈现的数据集上都体现除了强大而稳定的效果。最后，我们确定著名的多项式朴素贝叶斯分类方法在通常情况下比多变量伯努利朴素贝叶斯效果更好更稳定，进一步地，我们还确定使用二值化特征的多项式朴素贝叶斯方法比标准多项式方法效果更好。这篇论文的数据和代码都可以公开的获得，所有实验都可以得到重现。

具体方法

我们使用线性分类器的公式来表示我们的模型，对于第 k 个实例我们的预测公式是

$$y^{(k)} = sign\left(w^T x^{(k)} + b\right)$$

关于与该公式等价的概率公式的详细信息，可以在 1998 年 McCallum 和 Nigam 发表的“A comparison of event models for naïve bayes text classification”中查阅。

令$f^{(i)} \in \mathbb{R}^{|V|}$表示训练集中实例i的词频向量，实例i的标签为$y^{(i)} \in \{-1,1\}$。V表示特征的集合，$f_j^{(i)}$表示特征$V_i$训练集实例i中出现的次数。定义累积向量$p = \alpha + \sum_{i:y^i=1} f^{(i)}$以及$q = \alpha + \sum_{i:y^{(i)}=-1} f^{(i)}$，其中α表示平滑因子。最后定义 log-count ratio：

$$r = \log\left(\frac{p/||p||_1}{q/||q||_1}\right)$$

2.1 多项式朴素贝叶斯(MNB)

在多项式朴素贝叶斯中，$x^{(k)} = f^{(k)}, w = r$，$b = \log(N_+/N_-)$。N_+,N_-是训练集中正例和负例的数量。但是，正如 Metsis 在 2006 年“Spam filtering with naïve bayes – which naïve bayes ?”中提到的，我们发现使用二值化的$f^{(k)}$效果更好。另$x^{(k)} = \hat{f}^{(k)} = 1\{f^{(k)} > 0\}$，其中 1 在这里表示指示函数（注：即大括号中条件满足时为 1，否则为 0）。$\hat{p},\hat{q},\hat{r}$都重新使用$\hat{f}^{(i)}$向量而非词频向量$f^{(i)}$。

2.2 支持向量机(SVM)

对于支持向量机，$x^{(k)} = \hat{f}^{(k)}$，w,b均通过最小化

$$\mathrm{w}^{\mathrm{T}}w + C\sum_{i}\max\left(0, 1 - y^{(i)}\left(w^{T}\hat{f}^{(i)} + b\right)\right)^{2}$$

得到。我们发现使用 L2 正则 L2 损失的 SVM 效果最优而 L1-loss 则稳定性不佳。我们在这里使用了 LIBLINEAR 库。

2.3 使用 NB 特征的 SVM

另一方面，保持 SVM 方法不变，我们使用$\mathrm{x}^{(\mathrm{k})} = \tilde{f}^{(k)}$，其中$\tilde{f}^{(k)} = \hat{r} \circ \hat{f}^{(k)}$,∘表示向量间对应位置的元素相乘。上述方法在长文本上表现非常好，而且我们发现在 MNB 和 SVM 间做一个插值能使得其在所有文本上表现惊人。我们使用的模型如下：

$$\mathrm{w}' = (1-\beta)\bar{w} + \beta w$$

其中$\bar{w} = \left|\left|w\right|\right|_{1}/|V|$，表示w的一范数均值，β ∈ [0,1]是插值因子。这个插值可以认为是一种正则化：除非 SVM 非常有信心分类正确，否则相信朴素贝叶斯分类。

3. 数据集和任务

我们比较了在如下数据集中的公开结果。各数据集详细的信息统计见表附录 1-1.

表附录 1-1 各数据集详细信息.

Dataset	(N_+, N_-)	l	CV	$\lvert V\rvert$	Δ
RT-s	(5331,5331)	21	10	21K	0.8
CR	(2406,1366)	20	10	5713	1.3
MPQA	(3316,7308)	3	10	6299	0.8
Subj.	(5000,5000)	24	10	24K	0.8
RT-2k	(1000,1000)	787	10	51K	1.5
IMDB	(25k,25k)	231	N	392K	0.4
AthR	(799,628)	345	N	22K	2.9
XGraph	(980,973)	261	N	32K	1.8
BbCrypt	(992,995)	269	N	25K	0.5

RT-s: 短电影影评数据集，每个句子表示一条评论（来自 Pang 和 Lee）。
CR：顾客评价数据集(Hu 和 Liu)，与 Nakagawa 处理相似。
MPQA：MPQA 数据集上的子任务，包含极性观点。
Subj：包含主观观点和主观情节梗概的主观性数据集。
RT-2k：标准的包含 2000 条完整未截取的电影评价。
IMDB：包含 5 万条完整影评的数据集。
AthR，XGraph，BbCrypt：20-newsgroups 数据集中成对的新闻组数据，包含 alt.atheism 和 religion.misc ，comp.windows 和 comp.graphics 以及 rec.sport.baseball 和 sci.crypt，这些数据的新闻头部都被去掉了。

4. 实验和结果

4.1 实验初始化

对于数据的 Token 化，如果数据本身带有工具则使用其自带的，如果没有则将字符串按照空格分割，变为一个个单词，接着过滤掉单词中任何非[A-Za-z]字符。我们没有使用停用词表，词典或者其他资源。所有论文中的使用的参数，对于 NBSVM 均使用$\alpha = 1$，$C = 1$，$\beta = 0.25$,对 SVM 设置$C = 0.1$。

为了与其他公开结果比较，我们按照数据集的标准测试方法选择对应的或 10 折交叉验证或训练集、测试集的方式。交叉验证的折数使用表 1 的值，当原始数据集包含标准划分时则使用原始结果。对不同数据在$p < 0.05$的统计显著性近似上限见表 1 的Δ列。

4.2 MNB 在短文本上效果更好

Moilanen 和 Pulman 指出在每个实例均包含数百词的数据集上效果良好的统计方法，在短文本上效果并不好。往往基于规则的系统能较好的处理短文本问题。例子”not an inhumane monster “或者 “killing cancer”有效地支持了这个观点。这两个例子都是表达积极的情感，然而其中却全部使用了负面的词汇。

之前一些研究短文本分类的工作使用了预先定义的极性反转规则，并从依存树上学习复杂的模型。这些工作似乎前景大好，因为它们打败了静心设计的、基于规则的基准方法。但是，我们发现事实上一些朴素贝叶斯和支持向量机的变体可以比这些前沿方法做得更好，即便这些方法使用词典、反转规则和无监督预训练等多种手段。实验结果如表附录 1-2 所示。

我们的 SVM-uni 方法与 nakagawa 的 BoF-w/Rev 以及 Pang 和 Lee 使用了二阶核函数和额外特征的 BoWSVM 方法取得几乎一致的结果。而除了 MPQA 数据集之外，MNB 方法在所有数据集上都比 SVM 方法效果好。表二说明对于短文本，一个线性的 SVM 分类器是一个很弱的基准线。MNB（和 NBSVM）在短情感分类上表现更好，且经常优于其他公开的结果。因此，我们发现基于规则的系统在短文本数据集上有壁垒这一假设是不对的。MNB 在短文本的效果要优于在长文本上的效果。因而 Ng 和 Jordan 说明在训练数据少时朴素贝叶斯方法优于 SVM 或者逻辑回归，而我们则论证了 MNB 在短文本上同样也更好。它们的实验中，朴素贝叶斯

方法仅当训练数据实例少于 30-50 时才能比 SVM 方法效果好，而我们证明即使在包含 9K 实例的相当大的训练集上 MNB 方法同样优于 SVM。

表附录 1-2 在短文本数据集上的分类效果比较.

Method	RT-s	MPQA	CR	Subj.
MNB-uni	77.9	85.3	79.8	**92.6**
MNB-bi	**79.0**	**86.3**	80.0	**93.6**
SVM-uni	76.2	86.1	79.0	90.8
SVM-bi	77.7	**86.7**	80.8	91.7
NBSVM-uni	**78.1**	85.3	80.5	92.4
NBSVM-bi	**79.4**	**86.3**	**81.8**	**93.2**
RAE	76.8	85.7	–	–
RAE-pretrain	77.7	**86.4**	–	–
Voting-w/Rev.	63.1	81.7	74.2	–
Rule	62.9	81.8	74.3	–
BoF-noDic.	75.7	81.8	79.3	–
BoF-w/Rev.	76.4	84.1	**81.4**	–
Tree-CRF	77.3	86.1	**81.4**	–
BoWSVM	–	–	–	90.0

4.3 SVM 在长文本评论上效果更好

如表附录 1-1 所示，RT-2k 和 IMDB 数据集包含更长评论。对比 MNB 在短文本数据集上的极佳表现，其在长文本上且显得惨不忍睹。SVM 方法在这两个长文本情感分类数据集上表现比 MNB 好很多，但仍然较一些公开的结果差。但是，NBSVM 方法却超过或者接近上述的前沿方法，即是其中一些方法还使用了额外的数据。这个情感分类结果见表附录 1-3.

4.4 二元模型的收益要根据任务来定

二元模型特征在文本分类中不是是经常被使用（因此，通常使用的，是单个词组成的集合），这可能是因为其在主题文本分类上极其有限的实用性，如表附录 1-4 所示。这可能表明主题词的表示是孤立的。但是，在表附录 1-2 和表附录 1-3，添加二元模型总是能够提高效果，而且有时超过之前公布的结果。这大概说明情感分类从二元模型中获益很大，这可能是因为它们可以刻画变化的动词和名词。

表附录 1-3 在长文本数据集上的分类效果比较.

Our results	RT-2k	IMDB	Subj.
MNB-uni	83.45	83.55	**92.58**
MNB-bi	85.85	86.59	**93.56**
SVM-uni	86.25	86.95	90.84
SVM-bi	87.40	**89.16**	91.74
NBSVM-uni	87.80	88.29	92.40
NBSVM-bi	**89.45**	**91.22**	**93.18**
BoW (bnc)	85.45	87.8	87.77
BoW (bΔt′c)	85.8	88.23	85.65
LDA	66.7	67.42	66.65
Full+BoW	87.85	88.33	88.45
Full+Unlab'd+BoW	**88.9**	88.89	88.13
BoWSVM	87.15	–	90.00
Valence Shifter	86.2	–	–
tf.Δidf	88.1	–	–
Appr. Taxonomy	**90.20**	–	–
WRRBM	–	87.42	–
WRRBM + BoW(bnc)	–	**89.23**	–

表附录 1-4 在长文本数据集上的分类效果比较.

Method	AthR		XGraph		BbCrypt	
MNB-uni	85.0		90.0		**99.3**	
MNB-bi	**85.1**	+0.1	**91.2**	+1.2	99.4	+0.1
SVM-uni	82.6		85.1		98.3	
SVM-bi	83.7	+1.1	86.2	+0.9	97.7	−0.5
NBSVM-uni	**87.9**		**91.2**		99.7	
NBSVM-bi	**87.7**	−0.2	**90.7**	−0.5	99.5	−0.2
ActiveSVM	–		90		99	
DiscLDA	83		–		–	

4.5 NBSVM 是一个鲁棒性很高的方法

NBSVM 在短文本和长文本，情感、主题、主观分类上均表现良好，而且往往能够超过之前的一些成果。因此，NBSVM 似乎是一个恰当的、强劲的基准线，适合那些瞄准去击败词袋模型的复杂方法。

NBSVM 的一个缺点是有一个插值系数β。在长文本上当$\beta \in [\frac{1}{4}, 1]$时其表现几乎是不变的（变化值在 0.1%之内），而$\beta = \frac{1}{4}$时在短文本上平均比$\beta = 1$高 0.5% 。将β设置为$[\frac{1}{4}, \frac{1}{2}]$比设置为一些极值能够使 NBSVM 方法更加稳定。

4.6 其他结果

多变量伯努利朴素贝叶斯方法往往比多项式朴素贝叶斯效果差。只有在短文本分类，使用 unigram 时 BNB 效果才能与 MNB 有可比性。通常来说，BNB 相比 MNB 不稳定，且效果差 10%以上。因此，McCalum 和 Nigram 以 BNB 做参照来测试效果是靠不住的。

对于 MNB 和 NBSVM，使用二值化 MNB$\hat{f}$稍好于使用原始频数特征f。但这种差异在短文本上是微不足道的。

使用逻辑回归代替 SVM 能够得到相似的效果，而部分结果可以认为是更一般地认为是生成模型与判别模型的比较。