目 录

第一章 绪 论

1.1 研究工作的背景与意义

随着信息化进程的不断推进，我国互联网市场空前繁荣。从 WEB2.0 时代开始，到如今的“互联网 +”时代，互联网已经融入了人们的生活中，也对各行各业产生了深远的影响。根据中国互联网络信息中心发布的《第 40 次中国互联网络发展状况统计报告①》显示，截止到 2017 年 6 月，我国网民规模达到 7.51 亿，其中手机网民更是达到 7.24 亿，占总体网民的 96.3%，使用率排名前三的互联网应用分别是即时通信（92.1%）、网络新闻（83.1%）、搜索引擎（81.1%），使用率最高的三个 app 应用则是微信（84.3%）、QQ（65.8%）、微博（38.7%）。可以明显看出，人们使用互联网以及获取信息的方式，已经从以前的桌面台式电脑，转变为移动端的手机、IPAD 等掌上设备。

移动设备的盛行，使得人们发布和接收信息都更加方便，据统计，在 2012 年，微博用户已经增长到 3.68 亿，其中 69% 通过移动设备登陆，每天能产生 1.17 亿条微博。手机用户的大量增加，让互联网信息爆炸式增长，并且产生了大量碎片化的信息，如微博、QQ 说说、留言、商品评论等。据有关部门统计，互联网全体文本信息中，80% 以上属于内容较少的短文本信息。

如果能够有效分析这些短文本，对其进行精确的分类，可以方便用户有效的梳理这些浩如烟海的文本并掌握对自身有用的信息，商家也能够根据信息的分类提供更加优质的服务。例如，内容提供商，如新浪微博、知乎等，可以对其提供的信息进行分类，让用户快速获取自己感兴趣的某一类信息，同时商家也可以统计用户浏览过的信息类，精确投放用户感兴趣的广告，减少无关广告对用户体验的伤害；政府相关部门可以根据一段时间内大众发布的微博或朋友圈等信息，掌握当前的热门话题、集中关注点，监控当前的社会舆情；电子商务平台，如淘宝、京东等，可以利用情感分类技术，提取商品的正面评论与负面评论给用户，让用户能够更好的筛选与判断优质商品。

但是和传统文章相比，短文本过于短小（通常在 100 字以内，一般是一句话的长度），不能提供足够的词共现（word co-occurrence）或上下文，以至于很难从中提取出有效的文本特征[?]。因此，常规机器学习技术与文本分类算法很难直接应用在短文本之上。那么如何准确高效的对短文本进行分类，成为了互联网从业者与互联网技术学者所面临的一个重要难题，其突破也会具有重要的商业价值与

① http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201708/P020170807351923262153.pdf

使用价值。

1.2 国内外研究现状

随着互联网中短文本信息的增多，短文本分类领域得到了广泛的关注，越来越多的学者投入到短文本分类的研究之中。但由于短文本短小、信息分散的特性，传统基于词频、词共现的分类方法通常得不到较好的效果，比如贝叶斯方法（Naive Bayes）、最大熵模型（Maximum Entropy Model）、K-邻近算法（K Nearest Neighbors）以及支持向量机（Support Vector Machines，SVMs）。因此学者们开始尝试从其他方面来改进短文本分类算法，比如语义分析（sematic analysis）、半监督（semi-supervised）算法和集中模型（ensemble models）。

在语义分析方面，纽约大学的 Sarah Zelikovitz 等人[?] 通过隐含语义索引算法（Latent Semantic Index，LSI）对短文本的语义分析，将文本中的词映射到潜在语义空间，来捕获文本单词之间的相关性，提升分类效果；清华大学的 Chen 等人[?] 通过改进的隐含狄利克雷分布（Latent Dirichlet allocation，LDA）模型，将短文本中的单词与多个粒度的话题相关联，从而拓展短文本特征。

在半监督学习方面，Juan Manuel Cabrera 等人[?] 根据分布式词语表示算法（Distributional Term Representations，DTRs），用半监督的方式统计语料中的文档出现特征以及词语共现信息，形成每个词语的上下文信息，最后强化文本表示，以此来克服短文本处理中长度过短、信息高度分散的难点。

中国科学院自动化研究所的冯晓等人[?] 则构造了一种集中学习模型，直接确立短文本实例与某一主题直接的相关性，而不是将短文本表示为权重向量，取得了超过向量空间模型（Vector Space Model，VSM）的效果。

随着神经网络与深度学习逐渐兴起，并在计算机视觉、语音识别等领域取得了不错的成果，越来越多的学者注意到深度学习模型对于特征提取与数据建模上的优势，开始尝试在在自然语言处理问题中引入。而通过神经网络提取出的文本特征向量，可以直接用于其他任务，例如输入传统分类器进行分类。斯坦福大学的 Richard Socher 等人[?] 构造了一个使用矩阵向量的循环神经网络（MV-RNN），从长度不一致的句子中学习语义信息，形式长度统一的特征向量，最后放入分类器分类，取得了超过传统文本分类方法的结果；牛津大学的 Nal Kalchbrenner 等人[?] 提出了动态卷积神经网络（DCNN），利用动态 K-max 池化的方法，直接获取文本中单词直接的距离关系，避免了对语法分析树的依赖；哈佛大学的 Yoon Kim 等人[?] 将动态词向量与预训练好的静态词向量相结合作为同一段文本的两个表示，输入卷积神经网络的两个通道中进行分类，也取得的较好的效果。

但是，随着深度学习使用的逐渐扩大，学者们发现，对于语法复杂、需要分词的中文文本，深度学习模型并不能直接应用，也无法取得英文语料一样的优秀效果。因此，国内学者开始探寻适合中文文本的深度学习模型，也获得了很多不错的成果。

中国科学院自动化研究所的来斯惟等人[?]设计了一个循环卷积神经网络，在中文长文本分类任务中取得了较好的效果。纽约大学的张翔等人[?]实现了一个字符级别的卷积网络（ConvNets），将中文语句转化为汉语拼音，对中文语料进行分类。北京大学的李嫣然等人[?]通过在文本表示中引入部首信息，利用汉字中部首也包含一定语义信息的特点，在连续词袋模型（Continuous Bag-of-Words，CBOW）的基础上，构造出了一个新的字向量模型，在中文新闻标题分类上获得了明显的效果。

近几年，自然语言处理领域不断发展，一些新的方法也相继出现，2017 年，Google 公司开创性的提出注意力（Attention）观点，认为模型的结果并不是和每一个模型提取出的特征都有密切的关系，往往一个结果只是由某一个或某几个关键特征所决定的。这给了学者们新的启示，一些学者于是将注意力观点应用在文本分类之中。在文献 [**?**] 中，南洋理工大学的 Meng Joo Er 等人开发的基于注意力池化的卷积神经网络，利用平行的双向长短期记忆网络构造了一个输入文本的中间向量表示，以此作为卷积神经网络生成的文件特征向量的注意力权重，最后将经过处理后的文本特征向量输入分类器进行分类；卡耐基梅隆大学的 Zichao Yang 等人[?]则实现了一个分层注意力网络，整个网络由两层循环神经网络组成，第一层网络对句子进行建模，经过注意力机制处理之后得到句子的向量表示，然后第二层根据所有的句子向量得到文章的向量表示，最后根据这个文章向量获得分类结果。

尽管基于注意力机制的文本分类在近几年获得了研究人员的关注并得到了很多成果，但是这些成果大多是基于英文语料的，如何将该方法推广到其他语言中，并且如何通过注意力机制提升中文文本分类，特别是中文短文本的分类依然是研究人员需要继续的课题。

1.3 课题主要内容

1.4 本论文的结构安排

第二章 相关技术研究

随着互联网技术的发展与国民生活水平的提供，手机持有率也直线上升，这使得手机网民大规模的增长，人们在网上产生的信息、发布的文本日益碎片化。为了有效利用这些片段化的信息，短文本分类技术，作为自然语言处理的关键技术之一，也得到了充分的关注与发展，短文本分类应用的影子在互联网中随处可见。本章将对传统文本分类技术进行论述，并介绍本文涉及到的相关技术问题。

2.1 中文文本分词

词是最小的能够独立活动的有意义的语言成分，在英文文本中，每个单词天然的由空格分割开来，使得研究人员可以毫无难度的获取文本中所有的单词，然后进行接下来的研究工作。但对于中文这样由字组成的连续文本，不存在这样的语言优势，词语直接没有明确的区分标记，要想进行语义分析，必须先将中文文本切分，因此中文分词是中文信息处理的基础和关键。同时，大量实验表明，分词的好坏直接关系着后续分类算法的最终效果，所以选择一个快速并准确的分词算法尤为重要。目前常用的分词方法主要有两大类：基于字符串匹配的算法，基于规则的算法。

2.1.1 基于字符串匹配的分词算法

基于字符串匹配的分词算法又称为机械分词算法，主要依据外部提供的词典，按照一定的策略将待切分的中文文本与词典中的词条逐一匹配，若在词典中找到该词条，则匹配成功，否则做其它相应的处理。查找词库的匹配策略分为两种：长单词优先的最大匹配法以及短单词优先的最小匹配法。在实际使用中，人们发现单词切分次数越短，切分出的单词越长，分词效果越好，所以目前一般使用最大匹配法。按照匹配顺序的不同，最大匹配算法又可分为：正向最大匹配算法[?]、逆向最大匹配算法[?]、双向最大匹配算法[?]。

（1）正向最大匹配算法

正向最大匹配算法的基本思想是：已知词典中最长单词的长度为 N，则以 N 最为截取单词初始长度。对于带切分的中文文本 S，首先从左向右截取长度为 N 的字符串 W_1，然后在词典中寻找是否有和该字符串 W_1 匹配的单词。如果找到，则将 W_1 标记为成功切分出的单词，再继续从待切分文本的 $N+1$ 处开始下一次匹配；如果没找到，则将截取长度减 1 重新截取，即在 S 的原来位置重新截取长度为

$N-1$ 的字符串，然后重复之前匹配操作，直到截取长度为 1。算法流程如图2-1所示：

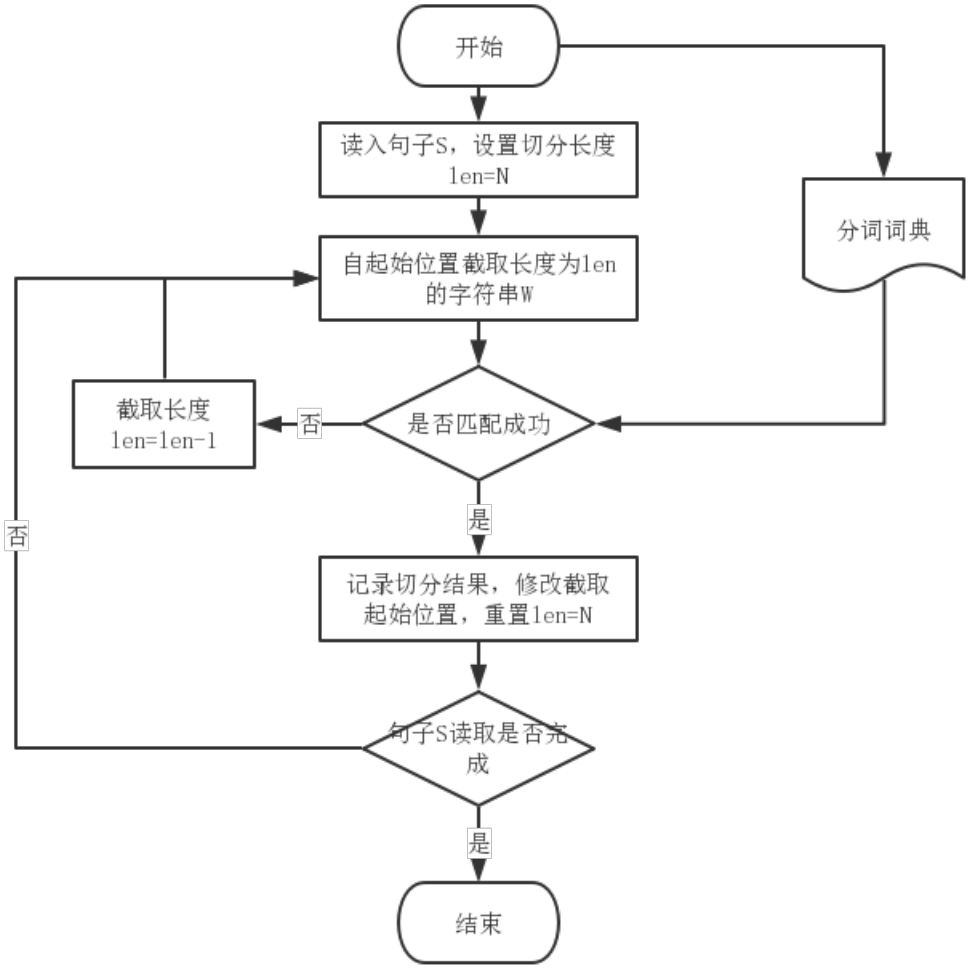

图 2-1 正向最大匹配算法流程

（2）逆向最大匹配算法

逆向最大匹配算法思路和正向最大匹配算法大致相同，不同之处在于截取字符串时的方向由从左向右换成了从右向左。也就是说，当对句子分词时，根据词典中最长单词的长度，从句子末尾开始向左截取字符串与词典中的单词匹配，直到切分到句子的开始位置为止。

（3）双向最大匹配算法

双向最大匹配算法是上述两种最大匹配算法的结合，侧重于分词过程中的检错和纠错，其基本思路是对待分词文本分别采用正向最大匹配和逆向最大匹配进行初步切分，然后将得到的正向分词结果和逆向分词结果进行比较，如果两种方

法的结果一致，则认为分词结果正确，如果结果存在出入，则认为分词存在着切分错误，需要采用其他技术手段消除结果中的歧义。

从上面的分析可以明显看出，无论哪种最大匹配算法都极度依赖词典，只有在词典覆盖的领域内，才有较理想的分词效果。对与词典没有覆盖的陌生领域语料，极端情况下甚至会出现单字切分的分词结果。

2.1.2 基于统计的分词算法

这类分词算法并不依赖具体的词典，而是根据语料中的统计信息，识别句子中的单词。即把单词看做是特定的字的结合，在语料中邻近的字共同出现的次数越多越可能是一个词。所以计算句子中特定字的组合的出现概率，可以判断这个组合是否是一个词。通过以概率论为理论基础，将中文文本中的每一个词的出现抽象成随机过程，分词算法不会被待分词文本的内容所影响，对所有领域的中文语料都有统一的效果，这是极度依赖词典的基于字符串匹配的分词算法所没有优势。根据采用的统计模型不同，基于统计的分词算法又可分为互信息算法、N 元统计模型等。

（1）互信息算法

在概率论和信息论中，两个随机变量的互信息是这两个变量彼此之间依赖性的一个度量。更确切的说，它是根据另一个随机变量来量化从一个随机变量中可以获得的信息量。互信息分词算法是互信息理论在分词中的应用，通过计算两个相邻字符串的互信息值，判断它们之间的结合程度。

对于两个相邻字符串 x 和 y，它们的互信息值计算公式如下：

$$I(x,y)=\log\frac{p(x,y)}{p(x)p(y)} \tag{2-1}$$

其中 $p(x,y)$ 表示字符串 x 和 y 在语料中共同出现的频率，$p(x)$ 与 $p(y)$ 分别表示字符串 x 与 y 的出现频率。当 $I(x,y)>0$ 时，表示 x 和 y 具有一定的相关性，并且这个值越大，它们联系的就越紧密，超过某一个阈值时即可判定为一个词；当 $I(x,y)=0$ 时，表示 x 和 y 的关系不明确；当 $I(x,y)<0$ 时，表示 x 和 y 直接几乎没有相关性，基本不会组成一个词。

（2）N 元统计模型

N 元统计模型又称为 N 元语言模型（n-gram language model），本质上是对语言建模的一种统计模型。该模型假定语言满足马尔科夫性，句子中的单词的出现与其前面出现的单词紧密相关，即第 n 个词的出现只与前面 $n-1$ 个词的出现相关，而和其他任何词都不相关。假设句子 S 由单词序列 $(w_1,w_2,...,w_m)$ 组成，则 N

元语言模型可表示为：

$$\begin{aligned} P(S) &= P(w_1 w_2 ... w_m) \\ &= P(w_1) P(w_2|w_1) ... P(w_i|w_{i-n+1} ... w_{i-1}) ... P(w_m|w_{m-n+1} ... w_{m-1}) \end{aligned} \tag{2-2}$$

理论上来说，N 取值越大，模型就越精确，越能揭示出语言的内在结构。但随着 N 的增加，模型的计算复杂度也呈几何式上升，所以在实际应用中，通常将 N 取值为 2、3、4，而 N 取 2 的 N 元统计模型称为 bigram 模型，N 取 3 的则称为 trigram 模型。

在分词应用中，算法首先对句子 S 进行全切分，得到若干分词结果，再根据公式2-2计算这些分词结果的概率 $P(S)$，最后选择概率最高的分词结果作为最终结果。

2.2 传统文本表示方法

分词处理之后，文本信息转化为一个单词序列，但对于计算机与程序来说依然是一段没有意义的字符串。为了让程序能够理解文本信息，继续之后的分类工作，我们需要对其再次处理，提取出其中的有效信息，将字符串文本映射成结构化的数字信息。

2.2.1 向量空间模型

向量空间模型（Vector Space Model，VSM）最早由 Salton 等人[?] 于 20 世纪 70 年代提出, 是一种将文本信息表示为向量的代数模型。模型的主要思想是将文本看做由单词的简单组合，通过统计语料中不同单词的个数，构建一个 n 维的向量，向量中每一个纬度都代表一个不同的单词，单词在文本中存在，则此位为 1，否则为 0，以此将一段文本信息转换为一个 n 维的数学向量。

可以明显看出，虽然向量空间模型建立了一个从文本到向量的快速转换，让程序能够容易地对文本进行处理计算，但这种映射方式太过简单。将文本表示为单词的组合，忽略了词语的位置关系以及词语之间的相互联系，而将相同单词都统计为一类，也忽略了一词多义的情形，让程序难以进行进一步的语义分析。而且，当统计的语料足够多后，模型产生的向量会拥有一个巨大的维度，造成后续计算的维度灾难。

为了解决上述问题，实际使用中，通常使用文本的关键词作为文本向量，而不是使用所有单词。因此，如何选择关键词就变得尤为重要。

2.2.2 TF-IDF 特征提取

TF-IDF（term frequency–inverse document frequency）是一种常用的关键词提取技术，它表明了一个词对于语料库中一份文本重要程度。算法的中心思想是：一个词的重要性与它在文本中出现的次数成正比，但同时也与它在整个语料库中出现的次数成反比，即如果某个词在一份文本中出现频率很高，同时在语料库中其他文本中出现频率很少，那么就认为这个词对于这份文本非常重要。

在 TF-IDF 中，TF（term frequency）代表词频，对于文本 j 的单词 i，它的 TF 值可以通过公式2-3计算。

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2-3}$$

其中 $n_{i,j}$ 表示单词 i 在文本 j 中出现的次数，$\sum_k n_{k,j}$ 表示文本 j 中所有单词的出现次数之和。

IDF（inverse document frequency）代表逆文档频率，是对一个词可以提供的信息量的一个度量，可以体现这个词在所有文档中是否重要。详细的计算方法如公式2-4所示。

$$IDF_{i,D} = \log \frac{|D|}{1 + |\{d \in D : t \in d\}|} \tag{2-4}$$

其中 $|D|$ 表示语料库中的文本总数，$|\{d \in D : t \in d\}|$ 表示包含单词 i 的文件数量。最后，根据公式2-3和2-4可以得到单词 i 的 TF-IDF 值，如公式2-5所示。

$$TF - IDF_{i,j} = TF_{i,j} \cdot IDF_{i,D} \tag{2-5}$$

2.3 传统文本分类方法

传统文本分类方法使用机器学习中的分类器对文本特征向量进行分类，这类分类器本质上是通过设定一个能够将任何特征向量映射到某一具体类别的理想目标函数 γ（$\gamma : D \to C$），然后根据学习算法减少自身的误差不断接近目标函数，最终实现分类的目的。按照设定的目标函数的不同，分类器可以分为线性分类器与非线性分类器，下面将对几种常见分类器作简要介绍。

2.3.1 贝叶斯分类器

朴素贝叶斯分类器[?]是一种典型的线性分类器，并且也是最古老及最简单的分类器之一。在自然语言处理领域有着广泛的应用，如垃圾邮件检测，个人邮件排序，文本分类，色情内容检测等。朴素贝叶斯理论是该分类器的基本理论，它

在分类任务中对输入数据有一个基本假设，即：数据的各特征之间是条件独立的。虽然这个假设看起来可能明显是错的，但依据朴素贝叶斯理论实现的朴素贝叶斯分类器在结构化或半结构化数据上都有比较理想的表现。同时，朴素贝叶斯分类器对 CPU 和内存的消耗也少于其他分类器。

朴素贝叶斯理论是指在上面提到的基本假设下，对于事件 A 和 B，它们之间的概率关系满足公式2-6。

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2-6}$$

而在文本分类任务中，假设一篇文本的特征向量为 $D = \{d_1, d_2, ..., d_m\}$，它的类别为 C，则上式可以写为公式2-7。

$$P(C|D) = \frac{P(D|C)P(C)}{P(D)} \tag{2-7}$$

其中 $P(C)$ 和 $P(D)$ 是先验概率（prior probability），分别表示类别 C 和特征向量 D 出现的概率，$P(D|C)$ 代表在类别 C 中出现特征向量 D 的概率，$P(C|D)$ 则是分类器的结果，称为后验概率（posterior probability），代表特征向量 D 是类别 C 的概率。实际应用中，$P(C)$ 和 $P(D)$ 以及 $P(D|C)$ 都可以根据训练语料库直接计算得到。

2.3.2 支持向量机

支持向量机（Support Vector Machine，SVM）是一种感知机的改进算法，在机器学习算法中有非常广泛的应用。

在感知机模型中，分类器通过寻找一个可以将数据正确分为两类的超平面来实现二元分类任务。但是，如图2-2所示，这样的超平面往往不是唯一，并且不同的超平面选择会导致感知机的分类准确率截然不同。

为了选择一个分类效果最佳的超平面，支持向量机将距离超平面最近的点设定为支持向量，然后让这些支持向量和超平面直接的距离最大，从而选择出一个最优的超平面，如图所示。并且对于线性分类不适用的非线性数据（如文本数据），支持向量机利用核函数的技巧，通过选择一个恰当的转换函数，将非线性数据映射到一个更高维的向量空间当中，让数据重新分布为线性可分的形式。但是这种做法增加了算法复杂度，复杂的核函数会使算法的计算量成倍的提高，而且如何选择合适的核函数让数据线性可分也没有一个统一的方法。

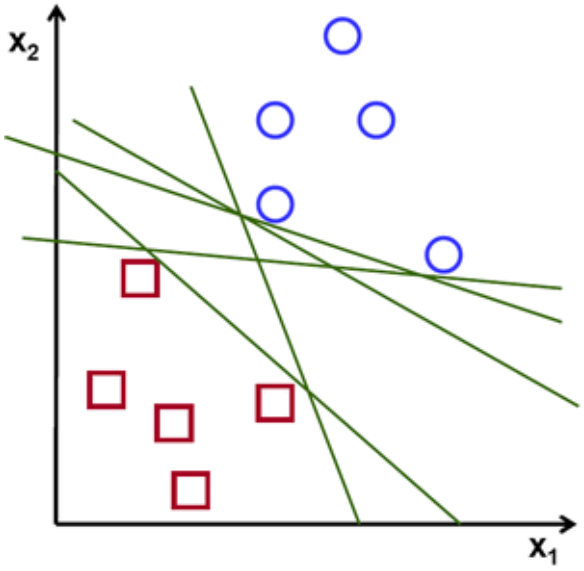

图 2-2 超平面

2.4 基于神经网络的短文本分类方法

传统文本分类方法虽然对于篇章级的长文本能够有较好的分类效果，但是对于单条文本信息量较少的短文本来说，特征提取后会形成大量的稀疏特征向量，数据只集中在向量中的某几维上，其他维度都为零。这就使得后续的分类算法难以发挥效用，为了突破这个难点，

2.4.1 词嵌入

2.4.2 卷积神经网络

2.4.3 循环神经网络

2.4.4 注意力机制