

## 目 录

<b>第一章 绪 论</b>	1
1.1 研究工作的背景与意义	1
1.2 国内外研究现状	2
1.3 课题主要内容	3
1.4 本论文的结构安排	3
<b>第二章 相关技术研究</b>	4
2.1 中文文本分词	4
2.1.1 基于字符串匹配的分词算法	4
2.1.2 基于统计的分词算法	6
2.2 传统文本表示方法	7
2.2.1 向量空间模型	7
2.2.2 TF-IDF 特征提取	8
2.3 传统文本分类方法	8
2.3.1 贝叶斯分类器	8
2.3.2 支持向量机	9
2.4 基于神经网络的短文本分类方法	10
2.4.1 短文本的分布式表示方法	10
2.4.2 卷积神经网络	13
2.4.3 循环神经网络	15
2.5 本章小结	17
<b>第三章 中文文本表示方法的研究和改进</b>	19
3.1 汉字偏旁部首的语义信息	19
3.2 中文文本表示方法	20
3.3 部首信息增强的中文词向量构建方法	21
3.3.1 文本预处理	21
3.3.2 部首信息增强的中文词向量模型	22
3.4 部首信息增强的中文字向量构建方法	25
3.4.1 中文字向量概述	25
3.4.2 部首信息增强的中文字向量模型	25
3.5 实验及其结果分析	25

参考文献 .....	26
------------	----



## 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<sup>①</sup>》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<sup>[1]</sup>。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/hlwxxzb/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 (semantic 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<sup>[2]</sup>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<sup>[3]</sup>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<sup>[4]</sup>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<sup>[5]</sup>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<sup>[6]</sup>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<sup>[7]</sup>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<sup>[8]</sup>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<sup>[9]</sup>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<sup>[10]</sup>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<sup>[11]</sup>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[12]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<sup>[13]</sup>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<sup>[14]</sup>、逆向最大匹配算法<sup>[15]</sup>、双向最大匹配算法<sup>[16]</sup>。

##### (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：

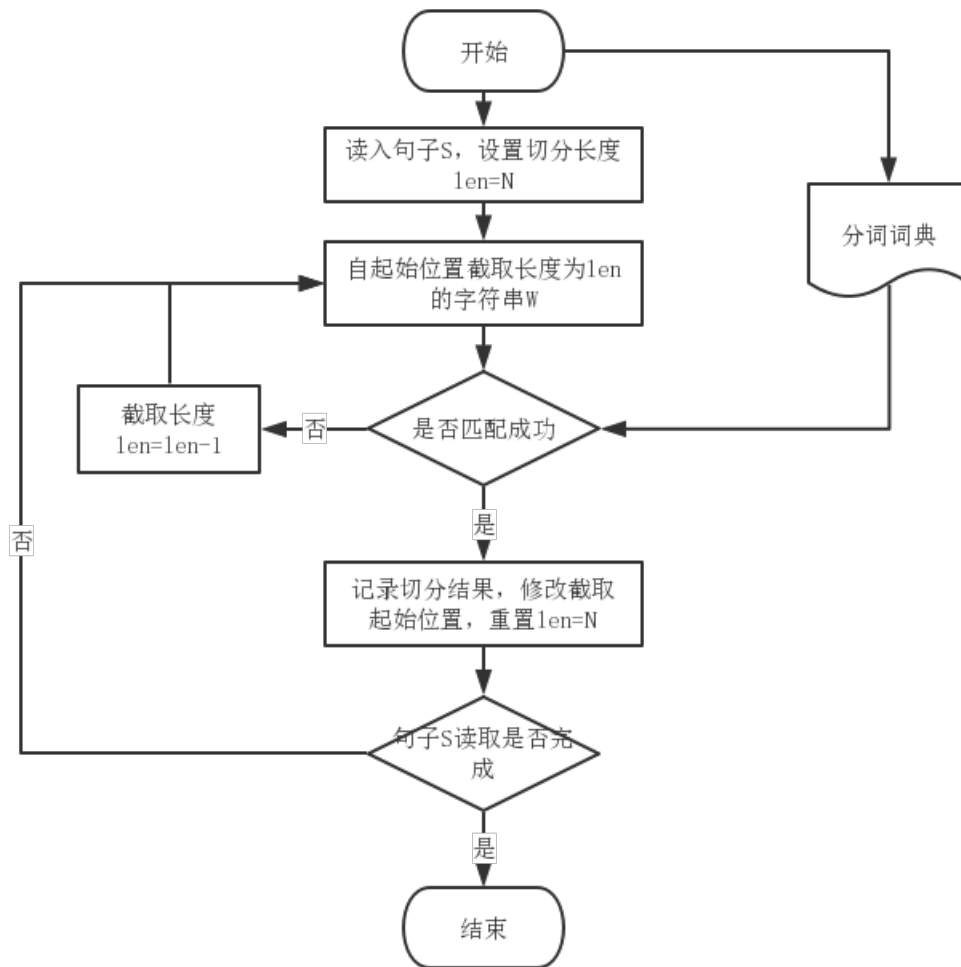


图 2-1 正向最大匹配算法流程

### (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：

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

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  $(w_1, w_2, \dots, w_m)$  组成，则 N

元语言模型可表示为：

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

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<sup>[17]</sup>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。

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

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。

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

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。

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

## 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  $\gamma$  ( $\gamma : D \rightarrow C$ ), 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<sup>[18]</sup>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。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: 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。

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

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

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

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。

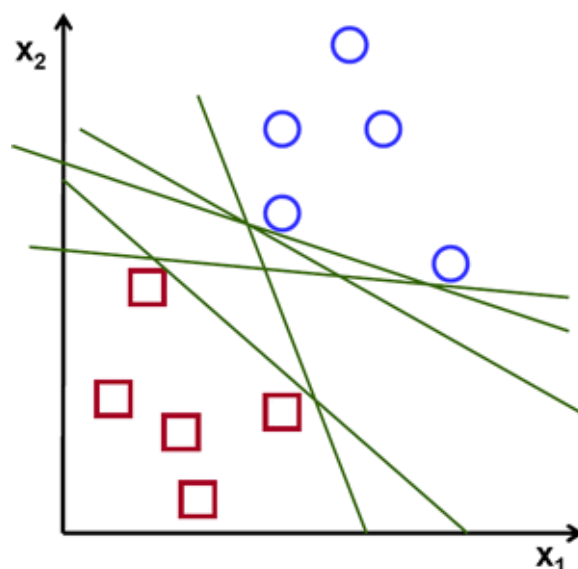


图 2-2 超平面

## 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在篇章级的长文本上能够有较好的分类效果，但是对于文本信息量较少的短文本来说，却难以发挥效用。基于统计的特征提取方法会在训练时生成庞大的特征词库，使得文本特征向量的维度依然很高，一旦文本长度过短，特征向量就会变为稀疏向量，即数据只集中在向量中的某几维上，其他维度都为零。文本特征向量的稀疏性一直是短文本分类上最难以解决的问题之一，为了解决这个难题，学者们开始尝试引入神经网络与深度学习技术。本小结主要介绍基于深度学习的文本表示方法，以及两种常见的深度学习网络。

### 2.4.1 短文本的分布式表示方法

短文本的分布式表示方法也称为词向量表示法。在神经网络与深度学习模型中，输入数据通常需要转化为矩阵的形式。对于自然语言处理任务，则需要将文本中的词语表示为一个向量，从而让模型理解文本并进一步提取特征，完成分类任务。

**One-hot Representation** 是词向量表示法最简单的实现方式，具体方法与前文中介绍的向量空间模型大致相同，针对具体语料可以快速得到每个单词的词向量。但与向量空间模型一样，也存在维度过高与稀疏性的问题，并且从词向量之中无法得到词语之间的关系，对短文本分类任务并不适用。

为了得到连续稠密的词向量，Hinton 等人提出了一种称之为词嵌入（Word Embedding）的词向量实现方法<sup>[19]</sup>。该方法旨在根据语料库中词语的分布式属性，来量化不同词语之间的语义相似度。通过一系列转换函数，将词语从高纬度文本

空间映射到一个维度相对较低的向量空间，表示为低维实数向量，从而有效的解决了向量的稀疏问题。并且这种映射方法还保留了词语之间的关系，即意思相近的词语它们产生的词向量也会在向量空间中彼此靠近。

目前最常用的词向量构建工具是 **word2vec**，由 Google 公司于 2013 年发布<sup>[20]</sup>。工具基于 2.1.2 中介绍的 N 元语言模型思想，假定文本中的一个单词只和其相邻的 N（word2vec 中称为窗口大小）个单词相关，构建了两套词向量模型——CBOW（Continuous Bag-Of-Words Model）模型与 Skip-gram 模型。

CBOW 模型仿照神经网络语言模型的思想（Neural Network Language Model, NNLM），根据当前词的上下文环境来预测当前词，如图 2-3 所示。不同的是，

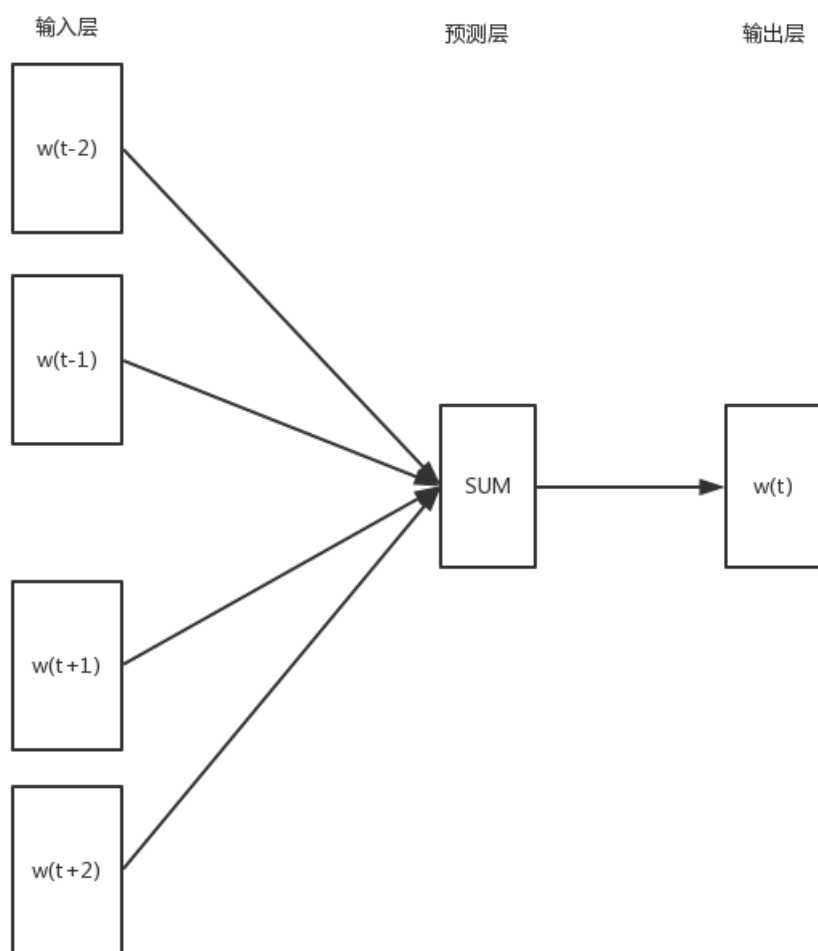


图 2-3 CBOW 模型

CBOW 删除了 NNLM 中前向反馈神经网络的隐藏层，将中间层直接和输出层的 softmax 节点连接，并且忽略当前词上下文的顺序信息，把输入的全部词向量都汇

聚在同一个隐藏层节点上。因此从数学上看，CBOW 模型相当于是把一个词袋模型的向量乘以一个嵌入矩阵，来得到一个连续的词嵌入向量。

与 CBOW 模型相反，Skip-gram 模型的思想是根据当前词预测与当前词相邻的词，模型结构如图2-4所示。对于句子  $S$ ，Skip-gram 模型的目标函数如2-8所示。

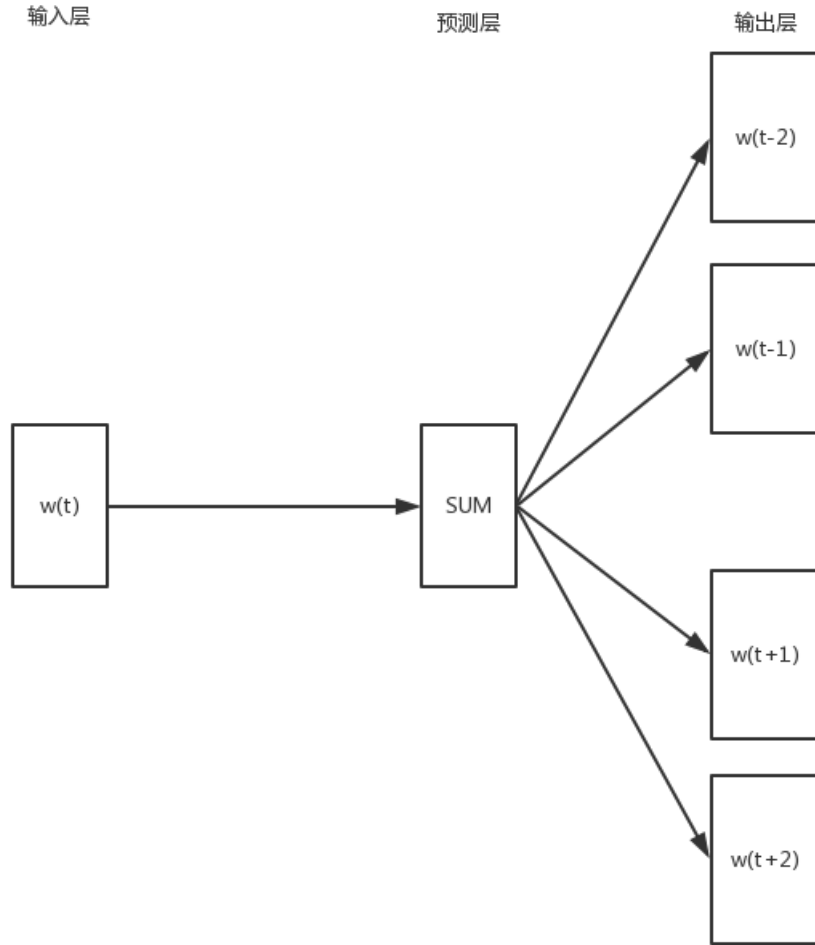


图 2-4 Skip-gram 模型

$$\arg \max_{\theta} \prod_{(w,c) \in S} p(c|w; \theta) \quad (2-8)$$

其中  $w$  为当前词， $c$  为当前词  $w$  周围的一些单词，长度由模型的窗口大小决定， $\theta$  是隐藏层参数。

word2vec 工具的优点主要体现在计算快速，通过简化传统 NNLM 的结构，引入分层 softmax (Hierarchical softmax) 和负采样 (Negative Sampling) 技术，极大的减少了预测层到输出层之间的计算量。同时模型采用非监督训练，适用于无标

签的大型语料库。

word2vec 工具的缺点也十分明显：针对英语的设计，只使用单词信息进行训练，使得模型不能够理解单词内在的一些语义信息。因此本文在原始 CBOW 模型之上，增加中文的字和部首信息，致力于实现一个适用于中文的词向量模型。

## 2.4.2 卷积神经网络

卷积神经网络 (Convolutional Neural Network, CNN) [21] 是一种前馈神经网络，通过模仿猫脑皮层中用于局部感知和方向选择神经元结构，有效降低了普通反馈神经网络的复杂性。

文本分类任务中，普通的神经网络模型通过构建一个全连接的网络对句子进行处理，输入层的神经元直接接收文本的词向量，计算时隐藏层的某些节点就会被输入的特征词激活（例如情感分类中“不”、“喜欢”等具有较强情感倾向的词），从而实现最终的分类。但是在这样的网络结构中，单词之间的位置信息被忽略了，而同样的特征词，顺序与位置不同很可能导致不同的分类结构。继续以情感分析为例，两个都包含“不”和“讨厌”这两个特征词的句子，“我不讨厌这部电影”表示的是好的情感，而“我讨厌这部电影，不会买它的票”则表示坏的情感。

虽然可以通过 n-gram 等关注上下文的多元语言模型来解决这个问题，但在实际训练中，模型的窗口大小会是一个非常大的问题。窗口过小会使得模型效果不佳，窗口过大则导致计算量爆炸。并且，相似的单词在模型中不能共享参数、权重，相似单词无法获得交互信息。

经过一系列探索，学者们发现卷积神经网络能够有效解决上述问题：基于局部感知野理论，一个隐藏层神经元只与部分词语连接，可以达到多元语言模型类似的效果，学习到更多的文本特征，并且在特征的识别与学习时只关注词语的相对位置，忽略它们的绝对位置，也加速了学习过程；基于参数共享理论，相似特征使用同样的参数来提取，极大的降低了网络模型的总体参数数量，加快了模型的训练。

卷积神经网络由三层网络组成：卷积层、池化层、光栅层，整体结构如图2-5所示。

### (1) 卷积层

卷积层主要用于特征提取，通过多个卷积核，对输入数据进行卷积计算，最后得到相应的特征图 (Feature Map)。

卷积计算是通信领域中常见的计算方法，假设有二维离散函数  $f(x, y)$  和



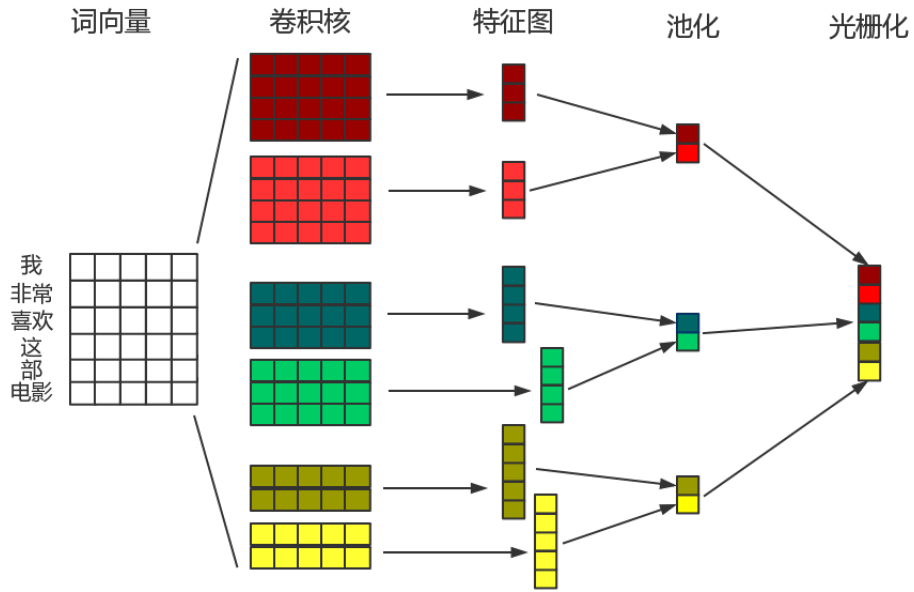


图 2-5 卷积神经网络结构

$g(x, y)$ ，那么它们的卷积定义如公式2-9所示。

$$f(x, y) * g(x, y) = \sum_u \sum_v f(u, v) g(x - u, y - v) \quad (2-9)$$

卷积核是提取特征的主要工具，相当于图像处理领域中的滤波器，可以从输入数据中提取一类特征，其他类特征则需要另外的卷积核来提取，所以卷积层通常包含有多个卷积核。一个卷积核包含参数矩阵  $W$  和偏置项  $b$ ，计算过程如公式2-10所示。

$$h = f(W * x + b) \quad (2-10)$$

其中  $h$  表示产生的特征图， $f$  是激活函数。

具体到文本分类任务，输入数据为文本的词向量，卷积核通常覆盖上下几行单词，并且宽度与词向量宽度相同（如图中卷积核部分所示），这样就能够捕捉到多个连续词之间的特征，并且能够在同一类特征计算时共享参数。

## (2) 池化层 (Pooling layer)

卷积层输出的特征图在数据纬度上与原始输入数据相比并没有减少很多，并且由于存在多个卷积核，实际上输出数据往往会增多，如果不做处理，无疑会增加网络的复杂度，产生巨大的计算。

池化层也称为下采样层，在卷积层之后，对特征图进行降维操作（即池化操

作)。常见的池化操作有最大池化、最小池化和平均池化。其中最大池化是现在最常见的池化方式，该方法将特征图分为多个子块，每个子块内部取最大的一个数据输出，从而达到降低特征图纬度的目的。

对于文本数据，池化操作可以将长度不同的句子转化为长度相同的特征向量，方便后续统一处理。另外最大池化还能够保留显著的特征。卷积核提取特征时，每一种卷积核会专注检测某一种含义的词组，如果这种类型的词组出现了，该卷积核此时的输出值就会非常大，通过最大池化就能够尽可能的将这些信息保留，同时忽略其他无关的信息。

### (3) 光栅层

输入数据经过卷积层与池化层计算之后，得到的是一系列特征图，而卷积神经网络之后的结构接收的输入可能是一个特征向量。因此需要将这些特征图中的数据依次取出，排列成一个向量，如网络结构图2-5中的光栅化部分所示。

## 2.4.3 循环神经网络

循环神经网络（Recurrent Neural Network，RNN）也叫做递归神经网络，是深度学习另一个重要的网络模型。RNN 的结构如图2-6所示，其特点是隐藏层节点不仅接收输入层的数据，还会接收上一个节点的输出数据作为另一个输入源，这让 RNN 可以非常自然的处理序列数据，如文本、语音、视频等。RNN 中隐藏层

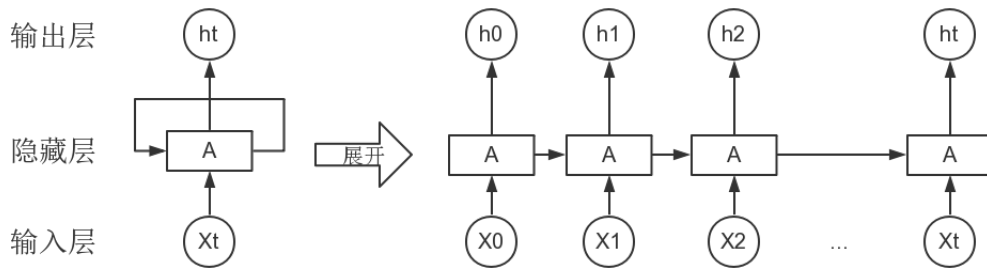


图 2-6 循环神经网络结构

节点包含三个部分，输入参数矩阵  $U$ 、记忆参数矩阵  $W$  以及输出矩阵  $V$ ，对于节点  $t$ ，计算公式如2-11所示。

$$s_t = f(Ws_{t-1} + Ux_t) \quad (2-11)$$

其中  $s_t$  为中间向量，同时送往下一个隐藏层节点  $t+1$  与输出节点  $h_t$ ， $f$  一般为非线性激活函数， $s_{t-1}$ 。输出节点通过公式2-12得到输出向量。

$$h_t = \varphi(Vs_t) \quad (2-12)$$

其中  $h_t$  是输出向量， $\varphi$  是输出函数，对于分类任务，一般为 *softmax* 函数。

RNN 模型的优点是对文本这类顺序数据有优秀的建模能力，能够突破 *n-gram* 模型中窗口大小的限制，提取长间隔的单词之间的特征，隐藏层最后一个节点的输出甚至可以包含前面所有单词的特征信息。同时模型长度可变，对任意长度的文本都可以直接处理。

RNN 模型的缺点也十分明显，就是模型训练十分困难。根据反向传播算法，RNN 模型训练时会将梯度在隐藏层之间以乘积的形式传播，如果梯度大于 1，则多次相乘会使其指数级上升，引起梯度爆炸（Gradient Exploding）；相反如果小于 1，都次相乘后梯度逐渐为 0，又引起梯度消失（Gradient Vanishing）。

长短期记忆网络（Long Short Term Memory networks, LSTM）<sup>[22]</sup> 是对传统 RNN 模型的改进，它将每个隐藏层节点看做“记忆细胞”，在里面增加一种称为“门”的结构，对输入输出数据加以控制，从而有效解决梯度爆炸与梯度消失的问题。

LSTM 的结构如图2-7所示，每个节点包含三个门结构（图中  $\sigma$  部分），分别为遗忘门（Forget Gate）、输入门（Input Gate）与输出门（Output Gate）。每个门由一个 *Sigmoid* 神经网络层和一个相乘操作组成，其中 *Sigmoid* 层输出 0 1 之间的值，表示对应的部分信息是否应该通过，并利用相乘操作作用在输入或输出信息上，0 值表示不允许信息通过，1 值表示让所有信息通过。在一次计算过程中，LSTM 节点首先计算遗忘门，公式如2-13所示。

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2-13)$$

其中  $W_f$  与  $x_t$  是遗忘门中 *Sigmoid* 层的参数矩阵和偏置矩阵， $f_t$  为遗忘向量，里面的每一位都对应  $C_{t-1}$  中的一个数据，用来控制信息的流入。

第二步是将输入数据  $x_t$  存储到节点中，并更新节点状态。这一步首先计算第二个门结构——输入门的 *Sigmoid* 层与一个 *tanh* 层，得到两个候选值  $i_t$  与  $\tilde{C}_t$ ，如公式2-14与2-15所示。

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2-14)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2-15)$$

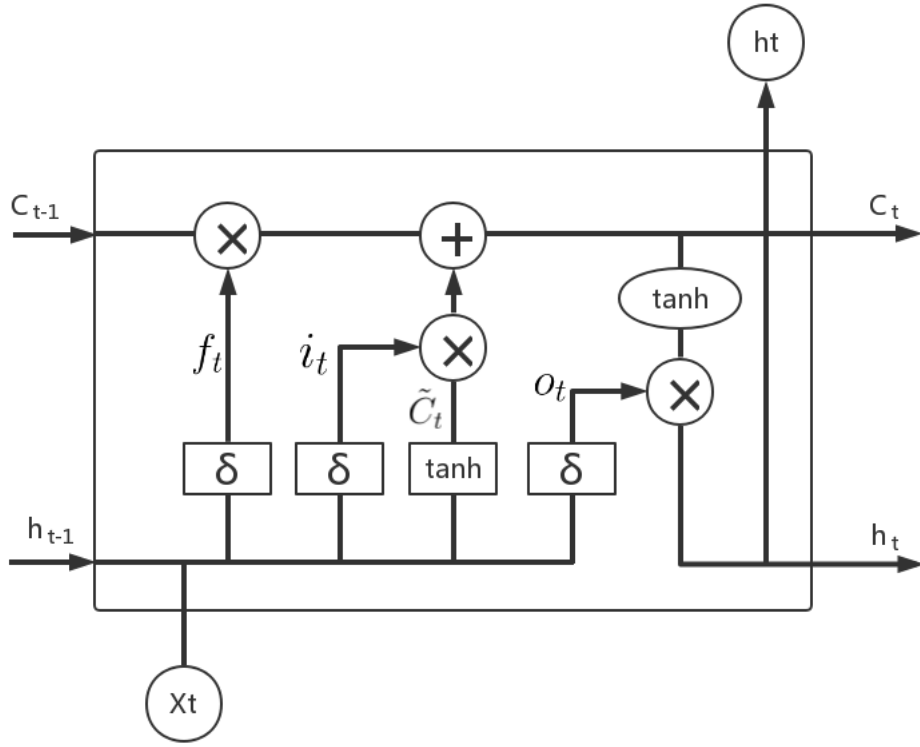


图 2-7 长短期记忆网络

然后根据公式2-16得到节点的新状态  $C_t$ 。

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2-16)$$

最后，节点通过第三个门结构——输出门，得到输出向量，如公式2-17所示。

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \quad (2-17)$$

通过增加门结构，LSTM 有效避免了传统 RNN 在训练上的问题，但与 CNN 相比，其训练效率依然十分低下，实际使用中常常造成问题。因此，本文对短文分类算法的优化，将融合 RNN 与 CNN，结合它们的优点，构建一个对于中文短文快速有效的分类模型。

## 2.5 本章小结

本章首先对传统的文本表示方法与文本分类算法的基本理论做了简要介绍，并指出了传统方法在中文短文本分类上的缺陷与不足。接着引出了基于神经网络

的短文本分类方法，详细介绍文本的词向量表示法以及卷积神经网络和循环神经网络在文本分类任务中的应用。这些基础内容为本文的后续工作提供了坚实的理论基础，后续章节将根据这些知识搭建一个中文短文本分类模型。

## 第三章 中文文本表示方法的研究和改进

文本的向量化表示是中文短文本分类研究的核心之一。传统的 One-hot 表示法虽然简单快速，但忽略的太多的语义信息，在实际使用中效果较差，达不到人们的预期。词嵌入技术将文本中的单词映射到一个连续的向量空间之中，让意思相近的单词能够在向量空间中彼此靠近，这让深度学习模型能够直接通过文本向量获取相关的语义信息，更有利与进一步的文本分类工作。因此，本章节将根据汉字的特点，充分利用汉字中蕴含的丰富信息，设计一种新颖的适合中文的词向量模型。

本章节首先介绍汉字中部首的一些特性，然后简要概述现阶段中文文本表示的一些方法，再详细介绍本文设计的基于部首信息的词向量与字向量模型，最后根据实验结果论述模型的有效性。

### 3.1 汉字偏旁部首的语义信息

汉字是四大古老文字之一，传说起源于黄帝的史官仓颉。根据内在结构的不同，汉字可分为独体字和合体字。独体字源于最早的甲骨文，是象形文字的延续。如图3-1所示，每一个独体字都可以看做是现实事物的抽象，本身不可分割，整体表达一个语义。合体字则是在独体字之上发展而来的。汉字系统初期几乎都为独体字，数量相对较少，大量事物以通假字来表示，使文字表述存在较大歧义。为了能更精确的表述，人们以基本的象形独体字为基础，通过组合的方式构造大量新的文字。比如最早海上的交通工具就只有：“舟”一种，但演化到现在，细分成“舢、舟、艇、船、舰”等不同小大规模与形制的“舟”。到了现代，合体字成为了汉字的主体，占 90% 以上。

部首是构建合体字的主要单元。这一概念最早由公元 2 世纪汉朝的许慎提出，他在其著作《说文解字》中将合体字中重复出现的部分加以归类，分成 540 个“部”。“部”之后发展为现在的部首，现在通用的部首由清朝康熙五十五年（1716 年）成书的《康熙字典》所制定，共 214 个。

汉字的部首通常由某一个独体字演化而来，而且与英语等字母语言不同的是，部首是汉字语义的重要组成部分。很多时候，部首能够让我们在任何上下文的情况下大致理解或推测一个汉字的意义。这也就是说，在学习汉字语义表示的时候，部首所固有的语义特征能够提供额外的信息。举例来说，通过由“人”字演变而来的部首“亻”，我们可以很清楚的知道“你”、“他”、“伙”、“侣”、“们”等字全部都和



图 3-1 鱼字的演化过程

人有关。并且这种根据部首获得的语义信息与 N-gram 模型这类基于相邻  $N$  个单词获得的信息本质上是不同的。因此在词向量模型中添加部首的语义特征能够给每个词语增加丰富的语义信息，提升中文词向量的质量。

偏旁是构建合体字的另外一个重要单元。虽然在合体字特别是形声字中，偏旁往往表示事物的读音，但是对于会意字与部分形式字，偏旁任然含有可利用的语义信息。比如汉字“武”，它的部首为“止”，是“趾”的本字，表示脚，偏旁为“戈”，表示武器，整体表示人拿着武器行走；对于汉字“茱”，部首的“艹”表示植物，偏旁“朱”则表示红色，兼表示字音。所以只要善加利用，偏旁也能够为词向量的构建提供很多语义特征。

### 3.2 中文文本表示方法

中文文本的词向量构建方法一直是中文自然语言处理领域的焦点。在研究初期，研究者们尝试直接使用英文词向量工具（如2.4.1中介绍的 CBOW 模型和 Skip-gram 模型）在经过分词之后的中文语料上训练词向量。但是这种做法很明显存在问题：大多数英文词向量工具在训练时都将词作为最小的操作单位，而忽略了词语内在的一些形态信息（morphological information）。和英语等字母文字不同，中文词语之中的汉字任然含有丰富的语义信息。比如“智能”这个词，它的语义信息一方面我们可以从语料中相关的上下文学习，就像 word2vec 模型一样。另一方面我们也可以根据组成这个词的两个汉字“智”和“能”推测出来（“智”为智慧，“能”为能力）。

为了解决上述问题，充分利用中文词语的语义信息，陈志勇等人<sup>[23]</sup>在普通 CBOW 模型上加入汉字信息，提出了 CWE 模型。随后 Xu 等人<sup>[24]</sup>在 CWE 模型

的基础上，赋予词中每个字权重，优化整个词向量模型。但这些模型都没有利用汉字中的偏旁部首，忽略了所有文字内部的信息。

另一方面，Sun 等人<sup>[25]</sup>在 CBOW 模型的基础上增加了部首信息来训练中文词向量；Yu 等人<sup>[26]</sup>在 CWE 模型中结合“部首-汉字”与“汉字-词”的信息，设计了一个多粒度的词向量模型。但这些方法都只是简单的在模型中添加了部首信息，没有考虑到部首的演化，即没有将部首与意义对应的汉字联系起来（比如“氵”-“水”），这使得模型从部首获得的信息非常有限，影响生成的词向量的质量。

### 3.3 部首信息增强的中文词向量构建方法

根据上一小节中介绍的其他学者在中文词向量模型上的种种成果以及汉字中偏旁部首的特性，本文设计了一种新的中文词向量模型——部首信息增强的中文词向量模型（Radical Enhanced Chinese Word Embedding Model, RECWE）。RECWE 以 word2vec 工具的 CBOW 模型为基础，在模型的预测层中增加一个代表“偏旁部首-汉字”信息的隐藏层，增强每个词的语义信息。同时，RECWE 将输入文本从简体转换为繁体，并对每个字的部首进行语义转换，最大程度上丰富输入词语的语义信息。

#### 3.3.1 文本预处理

由于实验所用的语料大多来自于互联网，其格式编码往往没有统一，所以需要首先进行相关处理，得到一个相对整齐一致的文本数据。同时为了让词向量模型能够获取部首信息，还需要进行语义相关的一些处理。整体的预处理流程包括：

##### （1）特殊字符过滤

网络搜集的数据常常是用户随意产生的非规范文本，里面会包含大量和文本语义无关的特殊字符与标点符号，因此需要对这些符号进行过滤，只保留文本以及相关英文专有名词。

##### （2）全半角转换

由于使用的语料来源广泛，相互之间没有共同的格式规范，文本中的数字、英文字母等同时存在全角和半角的格式，为了让之后的分词工作能够准确快速，需要将所有数字及字母统一转换为半角字符。

##### （3）分词

本文所使用的分词工具为 java 语言实现的 Ansj 中文分词开源库。该库基于 n-Gram、CRF 与 HMM 算法，分词速度达到每秒钟大约 200 万字左右，准确率能达到 96% 以上，并且实现了中文分词、中文姓名识别、用户自定义词典、关键字



提取、自动摘要、关键字标记等重要功能，对于短文本语料有较好的表现。

### (3) 繁简转换

新中国成立后进行了一系列汉字简化工作，现阶段中国大陆地区使用的简体汉字是 1956 年发布的<sup>[27]</sup>。但在汉字简化工作中，存在一些不适当的简化，使得简化字的部首与原始繁体字的部首不同，出现汉字语义上的偏差。例如繁体字中的“𧢲”字，分解后的偏旁部首为“𧢲”和“熊”，意指一种很大的熊，可是简化后却变成了“黑”，相应的偏旁部首为“罢”和“灬”，失去了原来的含义。而且简化工作还将多个繁体字归并为一个简体字，即“一简对多繁”，这就造成了理解上的困难和歧义，并生成了一批多音字，让简体词语中的字不能很好的反应这个词的语义，增加了词向量模型学习词语语义的难度。比如繁体的词语“干（gān）涉”、“乾（gān）燥”、“幹（gàn）部”中的“干”、“乾”和“幹”，简化后统一变为“干”字，让模型很难区分。

因此，为了更好的表达原始语料中文本的语义，让模型能够正确捕获中文词语中的部首的语义信息，需要将原本的简体文本全部转换为繁体文本。本文使用汉语言处理包 HanLP 作为繁简转换工具，该处理包能够有效区分同一个简体字对应的多个繁体字，并且可以识别简繁分歧词，如“打印机”与“印表機”。

### (4) 部首提取与语义转换

为了让 RECWE 模型能够直接获取汉字的部首信息，预处理阶段会将词语中的汉字拆解为对应的偏旁部首。但是，根据3.1节中关于部首的介绍，由某一个独体字演化而来的部首，在合体字中会为了书写美观往往需要进行一定的拉伸或收缩，来适应合体字整体的字型。例如“水”字在合体字中拉伸成了部首“氵”，“食”字在合体中收缩为了部首“饣”。因此本文提取了常见字的部首并构建了一个转换表（如表3-1所示），将部首转换为对应的独体字，从而让 RECWE 模型能够有效识别出这种内在联系，比如“木”字与“案”、“板”、“森”、“李”等字，“水”字与“泉”、“河”、“海”等字。

## 3.3.2 部首信息增强的中文词向量模型

本文提出的 RECWE 模型结合了中文的单词、汉字以及偏旁部首的语义信息，旨在构建适合中文的高效词向量。RECWE 模型基于 CBOW 模型<sup>[20]</sup>，通过词的平均上下文向量和字与部首的平均上下文向量预测目标词，并且使用这两个上下文向量的预测损失的和作为目标函数。

RECWE 模型结构如图3-2所示，其中  $w_i$  表示需要预测的目标词， $w_{i-1}$  和  $w_{i+1}$  分别表示文本中目标词  $w_i$  左边和右边的词。 $c_{i-1}$  和  $c_{i+1}$  表示  $w_{i-1}$  和  $w_{i+1}$  中的汉字， $r_{i-1}$  和  $r_{i+1}$  则分别表示  $c_{i-1}$  和  $c_{i+1}$  对应的部首（经过表3-1转换）， $s_{i-1}$  和  $s_{i+1}$  表示  $c_{i-1}$  和  $c_{i+1}$  的偏旁。通过在预测层将两个上下文向量并列以及共享隐藏层参

表 3-1 常见部首转换表

部首	对应汉字	部首	对应汉字
艹	艸	亻	人
刂	刀	犴	犬
灠	火	金	金
麥	麥	食	食
示	示	月	肉
攴	攴	𦉳	网
扌	手	氺	水
糸	糸	耂	老
牛	牛	忄	心
衤	衣	王	玉
辵	走	疒	病

数，字向量与偏旁部首向量在训练时能够很好的影响词向量，从而让最终训练得到的词向量能够充分反映出词语内部汉字与部首的语义信息。

与 CBOW 模型类似的，RECWE 模型的目标函数是两个上下文向量对于目标词  $w_i$  的条件概率的对数似然函数，如公式3-1所示。

$$L(w_i) = \sum_k^2 \log P(w_i|h_{i_k}) \quad (3-1)$$

其中  $h_{i_1}$  和  $h_{i_2}$  分别表示词的上下文向量和字与部首的上下文向量。通过这样

每个上下文向量对于目标词  $w_i$  的条件概率  $p(w_i|h_{i_k})$  可以利用 softmax 函数求解，如公式3-2所示。

$$p(w_i|h_{i_k}) = \frac{\exp(h_{i_k}^T \hat{v}_{w_i})}{\sum_{j=1}^N \exp(h_{i_k}^T \hat{v}_{w_j})}, k = 1, 2 \quad (3-2)$$

其中  $\hat{v}_{w_i}$  表示目标词  $w_i$  的“输出”向量， $\hat{v}_{w_j}$  表示输入语料中每个词的“输出”向量， $N$  表示输入语料的长度。

上下文向量  $h_{i_1}$  是上下文窗口中每个词的“输入”向量的平均值，通过式3-3得到：

$$h_{i_1} = \frac{1}{2T} \sum_{-T \leq j \leq T, j \neq 0} v_{w_{i+j}} \quad (3-3)$$

式中的  $T$  表示上下文窗口的大小， $v_{w_{i+j}}$  是上下文窗口中单词的“输入”向量。

类似的，字与部首的上下文向量  $h_{i_2}$  是上下文窗口中每个词对应的字及其偏旁

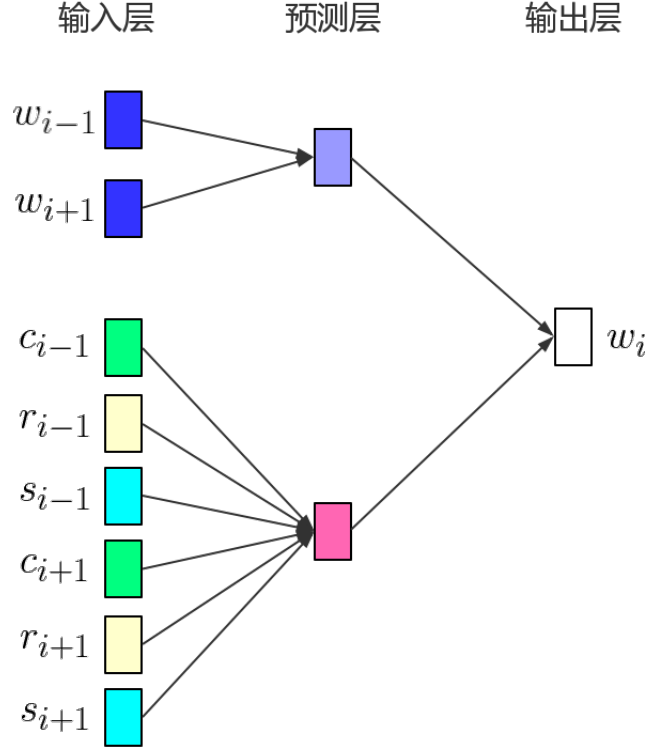


图 3-2 REWE 模型结构

部首的“输入”向量的平均值，计算公式如3-4所示。

$$h_{i2} = \frac{1}{X} \sum_{-T \leq j \leq T, j \neq 0} v_{c_{i+j}} + v_{r_{i+j}} \quad (3-4)$$

其中  $v_{c_{i+j}}$  和  $v_{r_{i+j}}$  分别表示字和偏旁部首的“输入”向量， $X$  为  $v_{c_{i+j}}$  和  $v_{r_{i+j}}$  的数量。

于是，对于语料库  $D$ ，RECWE 模型的整体对数似然函数如公式3-5所示。

$$L(D) = \sum_{w_i \in D} L(w_i) \quad (3-5)$$

RECWE 模型的训练算法采用 CBOW 模型实现的负采样 (Negative Sampling, NEG) 算法<sup>[20]</sup>，该算法是 NCE (Noise Contrastive Estimation, NCE) 算法的一个简化版本，其中心思想是将当前窗口内的词语分为正样本 (需要预测的目标词) 及负样本 (目标词之外的其他词)，然后加权随机选取负样本进行更新，每个负样本的权重与其在语料库中出现的次数成正比，以此来提高模型的训练速度并改善所得词向量的质量。结合随机梯度上升技术，基于负采样的 RECWE 模型能够快速获得中文词向量。

### 3.4 部首信息增强的中文字向量构建方法

#### 3.4.1 中文字向量概述

中文分词技术虽然已相对成熟，但准确率任然达不到 100%，分词结果依然存在一定的错误。对于本文主要研究的中文短文本，由于其大多来自与用户在互联网的互动，如腾讯空间说说、新浪微博、淘宝商品评价等，相比于长文本具有很强的随意性，充斥着大量的口语化表达方式，有些句子甚至存在语病，这更进一步增加了分词的难度，使得分词结果很不理想。而错误的分词结果会极大影响后续的词向量模型，降低词向量的质量。

字向量是解决分词问题的一个方向，来源于英文文本处理中字符级别 (Character-level) 的文本表示方式<sup>[10]</sup>。国内学者通过给汉字编码<sup>[23,28]</sup>，或将汉字转换为汉语拼音<sup>[10]</sup>，将文本表示为连续的向量，从而绕过分词的步骤，然后通过循环神经网络等深度学习技术直接从字向量中学习文本特征，取得了不错的成果。但这些方法得到字向量类似于 One-hot 向量，没有反应出每个字直接语义上的联系，忽略了大量的语义信息。因此，作为 RECWE 词向量模型的补充，本文设计了一个能够利用汉字部首信息的字向量模型，为后续的分类模型提供更多的文本信息，克服分词错误带来的影响，最终提升分类准确率。

#### 3.4.2 部首信息增强的中文字向量模型

### 3.5 实验及其结果分析

## 参考文献

- [1] G. Song, Y. Ye, X. Du, et al. Short text classification: A survey[J]. ISSN 1796-2048 Volume 9, Number 5, May 2014, 2014, 9(5): 635
- [2] S. Zelikovitz, F. Marquez. Transductive learning for short-text classification problems using latent semantic indexing[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2005, 19(02): 143-163
- [3] M. Chen, X. Jin, D. Shen. Short text classification improved by learning multi-granularity topics[C]. IJCAI, 2011, 1776-1781
- [4] J. M. Cabrera, H. J. Escalante, M. Montes-y Gómez. Distributional term representations for short-text categorization[C]. International Conference on Intelligent Text Processing and Computational Linguistics, 2013, 335-346
- [5] X. Feng, Y. Shen, C. Liu, et al. Chinese short text classification based on domain knowledge.[C]. IJCNLP, 2013, 859-863
- [6] R. Socher, B. Huval, C. D. Manning, et al. Semantic compositionality through recursive matrix-vector spaces[C]. Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, 2012, 1201-1211
- [7] P. Blunsom, E. Grefenstette, N. Kalchbrenner. A convolutional neural network for modelling sentences[C]. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, 655-665
- [8] Y. Chen. Convolutional neural network for sentence classification[C]. Proceedings of the 2014 Conference on EMNLP, 2014, 1746--1751
- [9] S. Lai, L. Xu, K. Liu, et al. Recurrent convolutional neural networks for text classification.[C]. AAAI, 2015, 2267-2273
- [10] X. Zhang, J. Zhao, Y. LeCun. Character-level convolutional networks for text classification[C]. Advances in neural information processing systems, 2015, 649-657
- [11] Y. Li, W. Li, F. Sun, et al. Component-enhanced chinese character embeddings[C]. Proceedings of the 2015 Conference on EMNLP, 2015, 829--834
- [12] M. J. Er, Y. Zhang, N. Wang, et al. Attention pooling-based convolutional neural network for sentence modelling[J]. Information Sciences, 2016, 373: 388-403

- [13] Z. Yang, D. Yang, C. Dyer, et al. Hierarchical attention networks for document classification.[C]. HLT-NAACL, 2016, 1480-1489
- [14] 张劲松, 袁健. 回溯正向匹配中文分词算法 [J]. 计算机工程与应用, 2009, 45(22): 132-134
- [15] 李娟, 周贤善. 一种改进的逆向匹配快速切分算法 [J]. 信息系统工程, 2010, 1(2): 133-134
- [16] 陈耀东, 王挺. 基于有向图的双向匹配分词算法及实现 [J]. 计算机应用, 2005, 25(06): 1442-1444
- [17] 薛苏琴, 牛永洁. 基于向量空间模型的中文文本相似度的研究 [J]. 电子设计工程, 2016, 24(10): 28-31
- [18] A. McCallum, K. Nigam, et al. A comparison of event models for naive bayes text classification[C]. AAAI-98 workshop on learning for text categorization, 1998, 41-48
- [19] G. E. Hinton. Learning distributed representations of concepts[C]. Proceedings of the eighth annual conference of the cognitive science society, 1986, 12
- [20] T. Mikolov, I. Sutskever, K. Chen, et al. Distributed representations of words and phrases and their compositionality[C]. Advances in neural information processing systems, 2013, 3111-3119
- [21] Y. LeCun, B. Boser, J. S. Denker, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation, 1989, 1(4): 541-551
- [22] S. Hochreiter, J. Schmidhuber. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780
- [23] X. Chen, L. Xu, Z. Liu, et al. Joint learning of character and word embeddings.[C]. IJCAI, 2015, 1236-1242
- [24] J. Xu, J. Liu, L. Zhang, et al. Improve chinese word embeddings by exploiting internal structure[C]. Proceedings of NAACL-HLT, 2016, 1041-1050
- [25] Y. Sun, L. Lin, N. Yang, et al. Radical-enhanced chinese character embedding[C]. International Conference on Neural Information Processing, 2014, 279-286
- [26] J. Yu, X. Jian, H. Xin, et al. Joint embeddings of chinese words, characters, and fine-grained sub-character components[C]. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, 286-291
- [27] 苏文正. 汉字简化问题研究 [D]. 兰州: 兰州大学, 2007, 53
- [28] 胡浩, 李平, 陈凯琪. 基于汉字固有属性的中文字向量方法研究 [J]. 中文信息学报, 2017, 31(3): 32-40