

**本 科 毕 业 设 计（论文）**



**题目: 基于深度学习的文本情感分类的研究与实现**

**姓 名 徐心仪**

**学 院 软件学院**

**专 业 软件工程**

**班 级 2013211502**

**学 号 2013212081**

**班内序号 32**

**指导教师 吴国仕**

**2017 年 5 月**

**北 京 邮 电 大 学**

**本科毕业设计（论文）诚信声明**

本人声明所呈交的毕业设计（论文），题目《基于深度学习的文本情感分类的研究与实现》是本人在指导教师的指导下，独立进行研究工作所取得的成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名： 日期：

**基于深度学习的文本情感分类的研究与实现**

**摘 要**

情感分析(Sentiment Analysis, SA)，也称为观点挖掘(Opinion/Review Mining)，指分析说话者所传达信息中隐含的情况状态，态度，意见等，在舆情分析，自动决策支持，广告推荐等很多方面都具有很大价值。文本情感分类是情感分析任务中的一种，主要识别给定文本的情感极性，由于自然语言文本的词汇丰富，又具有语义多样性，而且往往具有复杂的语法结构及不规范的语用现象如冗余，重复等，目前文本情感分类依然是一个具有挑战性的任务。本文主要研究如何使用深度学习完成语句级的文本情感分类任务，同时也实现了用于完成情感分类任务的知识模型以进行研究和对比。

本文的主要工作是构建了三个模型，首先是基于Stanford语法树的知识模型，其次，由于过于复杂的句式反而可能给神经网络带来干扰，本文提出了由C\&W模型改进，抛弃了全局语法特征的短语级卷积模型(Convolutional Neural Network on Phrase Level, CNNPL)。最后，由于CNNPL模型不能处理不定长的语句，也不能利用自然语言的序列性质，而单纯的循环神经网络如流行的LSTM模型不能提取文本的短语特征，因此本文结合CNNPL模型与LSTM模型，提出了短语级带有LSTM细胞的卷积模型(Convolutional Neural Network with LSTM on Phrase Level, CNNLSTMPL)。其中CNNPL模型和CNNLSTMPL模型均在NLPCC 2014的使用深度学习进行情感分析的任务集(SCDL)上取得了接近或略超过第一名的性能。

与此同时，本文提出了一套神经网络处理情感分类任务的流程，并对该流程中重要的部分进行参数分析。最后，本文采用Restful API框架实现了一套基于网页的展示系统。

**关键词** 深度学习 情感分析 长短期记忆模型 卷积神经网络

**Research and Implementation of Text Sentiment Analysis Based On Deep Learning**

**Abstract**

Sentiment Analysis, SA, also called Opinion/Review Mining, is aimed at analyzing the implied status, opinion, review of the speaker. It is widely used in many fields such as public opinion analysis, automated decision support, Ads recommendation. As a subtask, Text Sentiment Classification refers to detect the sentiment polarity of the input text. Due to the rich vocabulary, semantic ambiguity, complex grammar and irregular pragmatic phenomenon of natural language, Text Sentiment Classification remains a challenging task. In this paper, we concentrate on solving Text Sentiment Classification of the sentence level with deep learning method; however, we also implement a knowledge-based algorithm for study and contrast.

To solve this problem, we build three models: first, a knowledge based model using Stanford Parser; second, to get rid of the noise from complex grammar, we discard the global grammar feature and implement an model improved from C\&W model called Convolutional Neural Network on Phrase Level (CNNPL). In the last, because CNNPL cannot handle with variable length sequence, in the meanwhile, recurrent neural network such as Long Short-Term Memory (LSTM) cannot extract phrase feature, we implement a model called Convolutional Neural Network with LSTM on Phrase Level (CNNLSTMPL) which combines CNNPL and LSTM cell. Both CNNPL and CNNLSTMPL achieve good results in the task SCDL, NLPCC 2014. Their accuracies are close to or slightly exceed the best results on the contest of NLPCC 2014.

In the meanwhile, we prompt a process for handling sentiment classification using neural network, and then we analyze the important parameters of the process. Finally, we write a website to present those models by using Restful API.

**KEY WORDS** Deep Learning Sentiment Analysis Long Short-Term Memory Convolutional Neural Network

**目 录**

[第一章 绪论 1](#_Toc484568972)

[1.1 研究背景及意义 1](#_Toc484568973)

[1.2 主要挑战 1](#_Toc484568974)

[1.2.1 客观语句 1](#_Toc484568975)

[1.2.2 歧义 2](#_Toc484568976)

[1.2.3 实体 2](#_Toc484568977)

[1.2.4 标注集 2](#_Toc484568978)

[1.3 研究现状 2](#_Toc484568979)

[1.3.1 基于知识的情感分类方法 3](#_Toc484568980)

[1.3.2 基于传统机器学习的情感分类方法 3](#_Toc484568981)

[1.3.3 基于深度学习的情感分类模型 4](#_Toc484568982)

[1.4 论文主要研究内容及组织结构 5](#_Toc484568983)

[第二章 数据获取与预处理 7](#_Toc484568984)

[2.1 数据集 7](#_Toc484568985)

[2.2 情感词典 10](#_Toc484568986)

[2.2.1 中文情感词典 10](#_Toc484568987)

[2.2.2 英文情感词典 10](#_Toc484568988)

[2.3数据预处理 10](#_Toc484568989)

[2.3.1 基本流程 10](#_Toc484568990)

[2.3.2 分词 11](#_Toc484568991)

[2.3.3 输入特征 13](#_Toc484568992)

[2.3.4 稀有词删除 14](#_Toc484568993)

[2.3.5 英文词性还原及词干化 15](#_Toc484568994)

[2.3.6 特征生成详细流程 15](#_Toc484568995)

[第三章 基于知识的情感分析模型 22](#_Toc484568996)

[3.1 知识模型简介 22](#_Toc484568997)

[3.2 Stanford 语法树 22](#_Toc484568998)

[3.3 基本流程 24](#_Toc484568999)

[第四章 基于神经网络的情感分析模型 28](#_Toc484569000)

[4.1 深度学习相关算法 28](#_Toc484569001)

[4.1.1 人工神经网络 28](#_Toc484569002)

[4.1.2 反向传播网络 30](#_Toc484569003)

[4.1.3 优化方法 31](#_Toc484569004)

[4.1.4 卷积神经网络 33](#_Toc484569005)

[4.1.5 C&W模型 34](#_Toc484569006)

[4.1.6 循环神经网络 34](#_Toc484569007)

[4.1.7 LSTM 37](#_Toc484569008)

[4.2 CNNPL模型 38](#_Toc484569009)

[4.2.1 神经网络结构 39](#_Toc484569010)

[4.2.2 模型主要流程 39](#_Toc484569011)

[4.3 CNNLSTMPL模型 42](#_Toc484569012)

[4.3.1 神经网络结构 42](#_Toc484569013)

[4.3.2 模型主要流程 42](#_Toc484569014)

[第五章 结果分析 45](#_Toc484569015)

[5.1 模型分析 45](#_Toc484569016)

[5.1.1 实验条件 45](#_Toc484569017)

[5.1.2 精度分析 45](#_Toc484569018)

[5.1.3 时空复杂度分析 46](#_Toc484569019)

[5.2 参数分析 47](#_Toc484569020)

[5.2.1 初始编码策略 48](#_Toc484569021)

[5.2.2 优化函数选择 48](#_Toc484569022)

[5.2.3 激活函数选择 49](#_Toc484569023)

[5.2.4 是否过滤稀有词 50](#_Toc484569024)

[5.2.5 是否使用dropout 50](#_Toc484569025)

[5.2.6 是否使用词性 51](#_Toc484569026)

[5.2.7 word2vec编码训练方式 52](#_Toc484569027)

[第六章 情感分析系统 53](#_Toc484569028)

[6.1 整体模块 53](#_Toc484569029)

[6.2 开发环境 53](#_Toc484569030)

[第七章 结束语 55](#_Toc484569031)

[参考文献 57](#_Toc484569032)

[致 谢 63](#_Toc484569033)

[附 录 65](#_Toc484569034)

# 第一章 绪论

**1.1 研究背景及意义**

情感分析，又称为观念挖掘，指使用计算语言学的知识，对文本进行分析与自然语言处理，从而识别并抽取出语言材料中的主观信息的过程[1]。情感分析能够自动挖掘文本信息隐含的情况，态度，意见等，在舆情分析，自动决策支持，广告推荐等方面都具有很大价值[2]。目前情感分析的主要研究领域有情感分类，情感强度预测以及观点挖掘[1]。其中按照处理文本的粒度，情感分类任务常被划分为三种等级：一、篇章级，也即长文本，二、语句级，也即短文本，三、实体或特征级，四、单词或短语级。[2][3][4]。本文主要关注语句级情感分类任务，即抽取语句情感信息，判断整体情感极性。

目前国内外的研究学者和开发者主要通过三大类方法解决文本情感分类，即基于知识的方法，基于统计的方法或聚合的方法[5]。基于知识的代表性算法为基于情感词典的方法，例如Kim S M等[6]使用情感词典采用语义角色标注的方式衡量文本情感极性，Tong R M[7]则根据不同领域使用不同的人工词典。但基于知识的方法往往实现复杂，且鲁棒性一般较差，仅能在小规模的数据上取得一定成果，很难进行进一步的优化[5]。

基于统计的方法则主要为机器学习算法。由于情感分类本质上是文本分类问题，故典型的模式分类算法往往可以应用于情感分类领域，如Pang等[8]采用有监督的学习方法，包括朴素贝叶斯NB，最大熵ME和支持向量机SVM，获得了比人工标注的语料库方法更好的结果。

但传统的机器学习方法如NB，ME，SVM等基本可以归类为浅层学习，这些方法难以表达复杂的非线性函数，并过度依赖于研究人员依靠经验选择的高级特征。而随着神经网络这一技术的流行，由于其可以通过神经网络中多个非线性映射的隐藏层逐层计算逼近复杂函数，同时可以自动从低级特征中学到如何表达高级特征的特性，越来越多的研究者转向使用深度神经网络进行自然语言处理。目前，情感分类中仍存在许多问题没有得到很好的解决，因此，本文以提高文本情感分类系统性能为目标，重点研究如何使用深度学习对文本进行情感分析。本文实现的模型具有较好的迁移性，可以运用于评论分析，舆情分析等多个领域，因此，该课题具有较高的研究价值。

**1.2 主要挑战**

情感分类是一个非常热门的研究领域，自提出以来经过了井喷式的发展[3]。但实际分析中，待分析文本往往词汇丰富，又具有语义多样性，而且往往具有复杂的语法结构及不规范的语用现象如冗余，重复等[5]，并且还受到以下不可避免的问题约束，故该领域依然存在巨大挑战。

**1.2.1 客观语句**

定义客观语句为描述世界的客观现实的语句，而主观语句为表达个人感受，观点或信念的语句[3]，则一般认为主观语句是情感分类的主要研究对象。但实际上，主观语句可能并不带有情感信息，如“我认为蜜蜂蜇人更疼。”，而客观语句所描述的现实可能带有说话人的情感，比如“浴缸有条裂缝。”。后一句可能带有买家对浴缸这种商品的负向情感，故仅以主观语句作为分析对象无法获知说话人的真实情感。实际上，本文测试所用到的NLPCC 2014(The 3rd CCF Conference on Natural Language Processing & Chinese Computing)的SCDL(Sentiment Classification with Deep Learning) 数据集和展示用到的携程数据集都含有大量标注为带有情感极性的客观语句。

**1.2.2 歧义**

歧义消解一直是自然语言处理面临的核心问题，对于“你今天怎么这么早？”这句话，根据不同的上下文语境，人们可能有不同的理解。如：“你今天怎么这么早？怪不得教室被打扫地这么干净。”与“你今天怎么这么早？我还没做好派对的准备。”，在不同的上下文语境中表达了不同的情感。歧义消解不仅需要考虑上下文，还往往要求系统具备某一领域的常识。在上面例子中，“没有做好派对的准备”代表主人还不欢迎客人的到来，也即带有负向情感。

**1.2.3 实体**

在系统实际处理的文本中，说话人，即输入文本的作者常与不同实体如地点，机构，人名产生联系。例如“我买了美美的家具”，该句中的“美美”可能为正向形容词，也可能为“美美家居”。当将实体词误判为形容词或相反后，语句会存在歧义，其语法结构可能会被破坏，为情感分类系统及其它自然语言处理系统带来困难。

**1.2.4 标注集**

由于自然语言结构复杂，故采用机器学习训练自然语言时，往往需要大量标注集作为训练语料。因为采用人工方式进行标注需要大量的人力投入，标注集往往通过爬取评论等带有评分的语料文本自动获取标签。但本文通过研究发现，由于不同用户的打分标准不同，采用评分方式获取标签准确度不能达到百分之百，也即训练集和测试集中往往有错误。

以下是NLPCC 2014 SCDL数据集中的一条数据，在训练集中，该句被标注为带有正面情感倾向：

<review id="225">

不怎么怎么搞的，一带上去皮肤就过敏

</review>

实际上，即使人工标注数据，也往往存在标注规范不一致的情况。

例如对于“还可以吧”这句评论，说话者既有可能带有负向情感，也有可能带有正向情感。

**1.3 研究现状**

目前情感分析领域中基于统计的方法可以分为传统浅层机器学习方法和深度学习方法，因此本章将分别就基于知识的方法，浅层学习方法和深度学习方法在语句级任务上的研究进展加以介绍。

**1.3.1 基于知识的情感分类方法**

基于知识的情感分类方法往往利用可靠的情感词典，通过分析语句中较为常见的语法关系和语法现象对语句进行情感分类。

典型的构建情感词典的方式有人工构建和自动拓展种子词的方法，而后者又分为基于知识库的方法，基于语料库的方法和二者相结合的方法[9]。

基于知识库的方法一般需要有完备的语义知识库（如Wordnet）和人工事先标注正确情感的种子词，研究者依据种子词的情感极性，通过挖掘语义知识库中词语之间的关系推测未标注词的情感极性从而建立情感词典，如Hu[10]等通过同义词与反义词关系拓展情感形容词词典，最后人工筛选错误分类的词汇。由于知识库结构复杂，词汇经过若干次迭代后，可能会迭代为相反的情感极性，Kamps等[11]使用未标注词迭代为正向词和迭代为负向词所需次数之差来估计未标注词的情感极性。

由于情感词汇可能在不同的领域内表现不同的情感，而基于知识库的方法往往只能构建通用情感词典，因而适应性不强。基于语料库的方法则可以根据某领域内的大量语料推测情感词极性，能较好地解决该问题。Kanayama等[12]认为短句中连续的情感词极性相近，遇到转折词如but后，极性则会反转。Turney[13]使用词语与正面或负面种子词的点互信息(Pointwise mutual information, PMI)判断词语极性。二者结合的算法则能同时利用词义关系和语料中的共现信息，位置信息等关系，如Esuli[14]等使用PageRank算法，通过同义词间关系构建情感词典。与此同时，深度学习在构建情感词典方面的应用也逐渐增长，如Tang[15]等改进Word2Vec模型，通过在训练时添加中心词极性和所在句子训练带有情感的短语向量(Sentiment-Specific Phrase Embedding)。

典型的使用情感词典进行情感分类的算法可能是简单将情感词加和或结合语法特征分析情感。Choi等[16]定义一系列规则计算情感极性，例如对于二元词组，若第一个词为否定词，则其情感极性为第二个词的相反极性，若二者极性相同，则为第一个词的极性，否则，使用训练语料中词语的主要极性。Moilanen等[17]使用语法树计算语句情感极性，以此处理情感传播，极性冲突和极性转换等问题。Liang等[18]则通过依赖关系进行判断。

本文实现的基于知识的情感分类模型主要基于Stanford语法树。Baker等[19]于1979年提出使用隐马尔科夫模型训练基于概率上下文无关文法(PCFG)的语法树。但由于PCFG将短语中的每个部分等价处理，可能导致语法二义性，Ford等[20]提出了词汇化的PCFG（lexicalized-PCFG），该文法为每个规则添加规则的首要词HEAD，以获得语法树的更准确表示。Klein等[21]采用标记父节点和使用马尔科夫模型决定候选规则等方式优化了未词汇化的PCFG，该方法成为了Stanford语法树的理论基础。

**1.3.2 基于传统机器学习的情感分类方法**

有监督算法中，Pang等[8]采用有监督的学习方法，包括朴素贝叶斯NB，最大熵ME和支持向量机SVM，获得了比人工标注的语料库方法更好的结果。2004年,Pang等[22]使用图最小割算法进行情感分析。Nakagawa等[23]使用子依存树的极性作为输入特征之一训练带有隐藏变量的CRF模型来处理情感分类问题。Mullen等[24]使用包括PMI，Osgood语法差异[25]，主题相近程度和其它一些语法特征作为SVM输入。Kennedy等[26]将转折词等也作为输入特征。Abbasi等[27]提出了熵权重一般算法(Entropy Weighted Genetic Algorithm, EWGA)用以生成SVM的输入特征。

无监督方法中，Choi等[28]基于形容词在不同主题中情感极性可能不同这一特点，假设在同一主题中形容词的极性相同，提出了bootstrapping算法，该算法首先产生种子情感词，然后通过实体和情感词间的依赖关系识别训练样本中的主题将语句表达为特征向量，拓展种子词，最后通过Kmeans聚类获取语句情感分类。Lin等[29]基于LDA模型实现了LSM(Latent Sentiment Model)、JST(Joint Sentiment-Topic)、Reverse-JST(Reverse-Joint Sentiment-Topic)三种无监督的情感分析模型。

半监督算法中，Davidov等[30]通过高频词和内容词等从训练语料中抽取语句模式，删除商品描述中存在的模式，将语句和模式的匹配程度作为半监督学习器的输入特征。Riloff等[31]使用Meta-Bootstrapping和Basilisk算法进行主观名词的识别并以此获取语句主客观信息。Su等[32]提出了半监督最小割算法，该算法性能超过有监督最小割算法。

**1.3.3 基于深度学习的情感分类模型**

深度学习最初使用one-hot编码表示向量，输入特征空间复杂度至少为语句长度乘以词汇表大小，这很可能会引发维度灾难问题。Bengio等[33]根据单词上下文往往与单词自身语义有关的性质，通过训练带有一层投影层的神经网络，训练目标为用单词的上文预测单词本身来获取单词的向量表示。Mikolov等[34] [35]改进了Bengio的算法，取消了投影层，提出了Continuous Skip-Gram模型和 Continuous Bag-of-Words模型，并使用层次softmax或负采样方法训练模型。该模型产生的词向量编码被称为word2vec编码。Collobert等[36]提出了C&W模型。该模型主要分为输入层，查表层，卷积层，池化层，线性层等，被设计可应用于多种自然语言处理任务，包括对单词和语句的处理及分类，语法角色标注，词性标注等。C&W模型也可以运用于无监督或有监督生成词向量。Tang等[37]在C&W基础上，加入语句情感极性训练以此得到带有情感的词向量。

在使用深度学习进行情感分析方面，Socher[38]使用无监督递归自编码器(RAE)预测语句语法结构与每个节点的向量表示，加入softmax层用以分类文本。但Socher认为基于单个单词的RAE无法识别短语的位置信息，而计算短语的组合函数依赖于短语中的每个单词，因此Socher在2012年[39]提出了Matrix-Vector RNN模型以改进该模型。MV-RNN模型在Socher RAE模型的基础上，为每个单词添加参数矩阵（此处d为单词表长度），用于根据上下文计算短语特征。但也正是因为该模型为每个单词添加的参数矩阵导致算法总体时空复杂度过高，需要训练的参数也过多，Socher在2013年对该模型提出了改进[40]，对所有节点使用一个统一的基于张量的组合函数，即为RNTN(Recursive Nerual Tensor Network)模型。Bespalov等[41]使用滑动窗口建立n元短语表示模型并以此进行情感分析。Cao[42]结合CNN模型与SVM模型，将CNN输出作为SVM的输入特征进行训练，取得了不错的成果。

**1.4 论文主要研究内容及组织结构**

本文探索基于深度学习的文本情感分析方法，并针对文本情感分析的方法对模型进行改进和调整以适应特定任务。

本文主要实现了三个模型： 基于Stanford Parser语法树的纯知识模型，由C&W模型特化，抛弃了全局语法特征的短语级卷积模型(Convolutional Neural Network on Phrase Level, CNNPL)和结合CNNPL模型与LSTM模型的短语级带有LSTM细胞的卷积模型(Convolutional Neural Network with LSTM on Phrase Level, CNNLSTMPL)。

在三个模型中CNNPL模型和CNNLSTMPL模型在数据集上都表现良好，在NLPCC2014的SCDL数据集的中英双语上都接近或略超过比赛评测第一名的结果。在NLPCC 2014的SCDL数据集和用于展示的携程数据集上，CNNLSTMPL模型的精度都高于CNNPL模型，结果说明相较于CNN，RNN更适应于处理序列文本。同时，经过重复实验发现，相较于对整句抽取，对语句局部抽取特征后进行全局最大池化往往准确度更高，泛化能力更强。

本文组织架构如下：

第一章、绪论：介绍了本文的研究背景和意义，文本情感分类所面临的挑战，以及相关研究工作。同时介绍了本文的主要内容。

第二章、数据获取与预处理：介绍了测试集及对应获取方法，以及基于Stanford语法树的知识模型所用到的情感词典。分别介绍了中英文下文本情感分类的一般预处理流程和流程中用到的工具。

第三章、基于知识的情感分类模型：介绍了Stanford语法树的基础知识，以及知识模型的实现思路和方法。

第四章、基于神经网络的情感分类模型：介绍了模型中所用到的深度学习的算法，同时分别介绍了CNNPL模型和CNNLSTMPL模型的基本结构。

第五章、结果分析：使用不同数据集对三种模型进行性能分析。本章对深度学习模型参数进行调整分析，同时提出了若干适应于文本情感分析的深度神经网络的技巧。

第六章、情感分析系统：本文结合各情感分析模型，实现了基于网页的评论情感分类系统。本章主要介绍系统的模块及实现方法。

第七章、结束语：总结了整个研究过程中的经验，对系统的现有问题进行了归纳，对将来可以改进的地方做了总结。

**第二章 数据获取与预处理**

**2.1 数据集**

为了评估数据模型的准确性以及训练模型以便展示等目的，本文针对语句级任务获取了以下数据集。

表2-1 NLPCC2014SCDL中文数据集基本情况

|  |  |
| --- | --- |
| 名称 | NLPCC 2014 SCDL数据集中文 |
| 数据总数 | 12500条 |
| 数据集大小 | 4.4MB |
| 官方测试集数据总数 | 2500条 |
| 语言 | 中文 |
| 比例 | 正向:负向 6250:6250 |
| 测试集比例 | 正向:负向 1250:1250 |
| 词数最大值 | 800词数量级 |
| 99%词数 | 200词数量级 |
| 情感分类准确度 | 不够准确 |
| 语句特点 | 语句较为口语化，有广告等中性语句 |
| 获取方式 | <http://tcci.ccf.org.cn/conference/2014/pages/page04_sam.html> (Accessed at: 5/21/2017) |
| 主要用途 | 评估模型性能 |

表2-2 NLPCC2014SCDL英文数据集基本情况

|  |  |
| --- | --- |
| 名称 | NLPCC 2014 SCDL数据集英文 |
| 数据总数 | 12485条 |
| 数据集大小 | 9.0MB |
| 官方测试集数据总数 | 2500条 |
| 语言 | 中文 |
| 比例 | 正向:负向 6237:6248 |
| 测试集比例 | 正向:负向 1250:1250 |
| 词数最大值 | 6000词数量级 |
| 99%词数 | 1000词数量级 |
| 情感分类准确度 | 较为准确 |
| 语句特点 | 词汇较丰富，部分评论中单词没有以空格或其它分隔符分开 |
| 获取方式 | <http://tcci.ccf.org.cn/conference/2014/pages/page04_sam.html> (Accessed at: 5/21/2017) |
| 主要用途 | 评估模型性能 |

表2-3携程数据集基本情况

|  |  |
| --- | --- |
| 名称 | 携程数据集 |
| 数据总数 | 1710000条 |
| 数据集大小 | 39.2MB |
| 测试方法 | 5 fold |
| 语言 | 中文 |
| 比例 | 正向:中性:负向=57000：57000：57000 |
| 词数最大值 | 1750词数量级 |
| 99%词数 | 250词数量级 |
| 情感分类准确度 | 不够准确 |
| 语句特点 | 语句较为口语化 |
| 获取方式 | 通过selenium模拟浏览器爬取 |
| 主要用途 | 模型展示 |

**备注1 99%词数**

如图2-1，图2-2，图2-3，本文中数据集普遍存在少量过长评论，这些过长评论的词数往往是数据集中99%的语句的词数的4-8倍，但数量上却只占数据集的较小部分。由于CNNPL网络只能处理固定长度的语句，不足的长度需要用零补齐，在实际训练CNNPL网络的过程中，为这些过长的语句额外训练参数会大幅增加模型的时空复杂度，因此本文选取99%语句的词数作为CNNPL网络处理的词数，超过该词数的文本将被截断。同时，该措施也是为了训练CNNPL网络仅使用文本前半段判断文本整体情感极性的能力，以使CNNPL网络对变长文本具有一定处理能力。

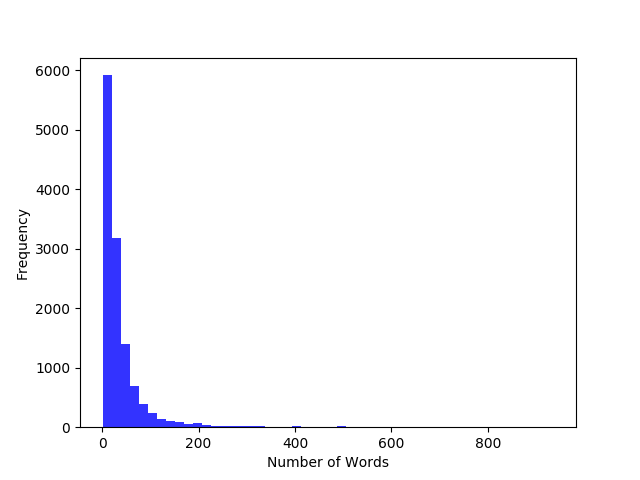


图2-1 NLPCC2014SCDL中文数据集词频分布图

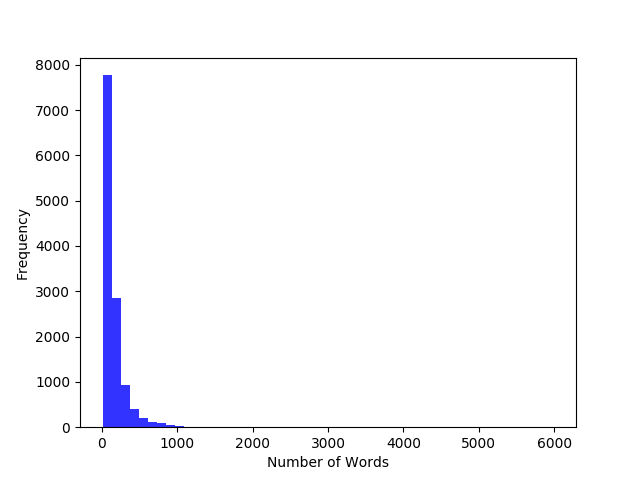


图2-2 NLPCC2014SCDL英文数据集词频分布图

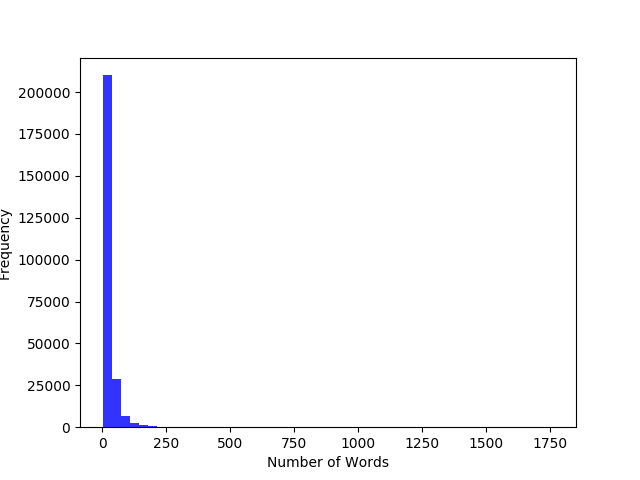


图2-3 携程数据集词频分布图

**备注2携程数据集获取过程**

由于NLPCC 2014的SCDL数据集样本容量较少，本文通过selenium模拟浏览器爬取携程网站中国内酒店带有评分的评论388067条用于展示和评估。由于携程酒店的评分是5分制，根据人们在购物网站评分时的心态，我们将满分5分的评论定为正向评价，严格小于4分的评论定为负向评价，其余的定为中性评价，得到正向评价193138条，中性评价137604条，负向评价57325条。为了平衡数据集，避免神经网络模型对特定标签产生偏置，我们随机抽取其中正向评价，中性评价，负向评价各57000条，组成携程数据集。

**2.2 情感词典**

由于知识模型可以同时支持中/英文分析，故情感词典主要由以下两种语言共六个词典组成。

**2.2.1 中文情感词典**

1. Hownet（知网）情感词典[43]：语言为中英双语。该词典为董振东和董强建立的情感分析用语集，包括主张词语，正面/负面情感词语，正面/负面评价词语，程度级别词语，并为程度级别词语划分强度等级。但其中有一些不常使用的词语，如"噲"，"媢"，"媢嫉"，"忺"，"安"，"巴"等在正面情感词典中出现的词。另外词语可能以短语形式出现，如"越...越..."，"abandon oneself to despaire"等。
2. NTUSD台湾大学情感词典[44]：台湾大学自然语言处理实验室提供的情感词典，包括正面词汇2810个和负面词汇8276个，极性划分较为准确，且包含各种词性，如“一下子爆发”，“一下子爆发的一连串，“一巴掌”。
3. DUTIR情感词汇本体库[45]：大连理工大学信息检索研究室整理和标注的中文情感词库，具有词性，情感分类，强度，极性和辅助情感分类，强度，极性等特征，划分详细。包含各种词性，但以成语和俗语为主体。

**2.2.2 英文情感词典**

1. SentiWordNet[46] [47]：基于WordNet3.0，具有语义，正向评分PosScore和负向评分NegScore等信息，本文中取PosScore-NegScore之差作为其分数。
2. Opinion Lexicon:Bing Liu等[10] [48] 整理的极性情感词典，仅包含单词本身，不包含短语，但包括单词的各种变形形式。
3. 匹兹堡大学MPQA主观性词典[49]：是MPQA(Multi-Perspective Question Answering，多方面问答)系统所用到的词典，具有情绪词，词性，情感强弱等词汇信息。

**2.3数据预处理**

在有监督学习的过程中，如何选择特征，选择何种特征将对训练结果和泛化性产生极大的影响。实际环境中，文本往往含有不规范的表达方式及符号，因此有必要对文本进行一系列预处理。本文根据现有文献的经验，分析目前流行的文本预处理工具后，根据不同语言选择不同预处理工具，得出了下列文本预处理的流程。

**2.3.1 基本流程**

图2-4表示了整个文本情感分类系统的基本预处理流程。对所有模型，本文都需要对文本进行转换编码，去除非法字符，化繁为简和短句切分，分词这些预处理工作。其中分词则是重中之重，会对所有模型的准确率产生很大影响。同时，本文的神经网络模型在训练模式下需要生成词汇表，训练待编码特征对应的编码，并将语句各特征抽取出来以备训练或预测情感极性使用。

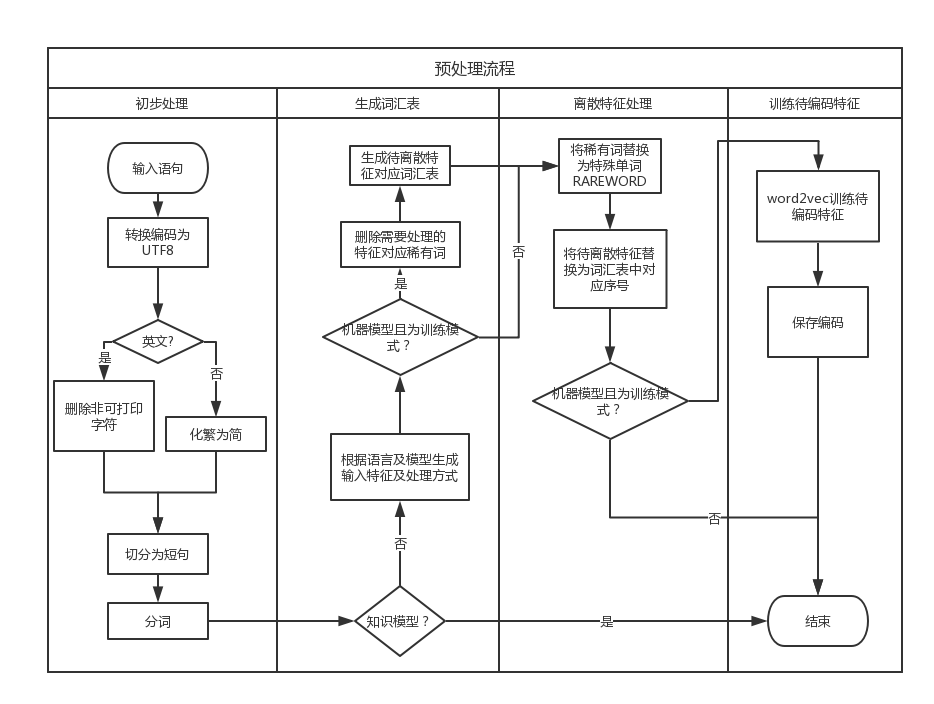


图2-4 数据预处理基本流程图

**2.3.2 分词**

**i.中文**

本文从携程数据集和NLPCC中文任务中随机抽取15条语句，使用相同的用户自定义词典，对比了三种分词工具thulac,ansj,jieba的精准分词性能，结果如附录1 分词工具准确率测试所示。由于ansj工具的实体识别能力最强，故选择ansj工具。

ansj是由孙健实现的，具有分词，标注词性，实体识别，关键词抽取及新词发现功能的java中文分词工具。该工具使用隐马尔科夫模型进行语义消歧，使用Hash和高度优化Trie树进行词典匹配，并使用条件随机场模型(CRF)进行新词发现[50]。本文主要使用到其分词，标注词性，实体识别及新词发现功能。

对于新词发现功能，本文主要学习输入语料中的新词，然后对这些新词进行手工过滤，加入情感词典以及ansj用户自定义词典。其中用户自定义词典人工过滤8430条，过滤得到3137条词条，157条情感词典词条。

**ii.英文**

与没有分隔符的中文相比，英文短句中的单词之间由空格作为分隔符，更易于分隔，因而英文的分词难度远远小于中文分词。但由于英文文本中存在标点符号和不规则语用现象，本文选择使用Stanford CoreNLP工具进行分词。同时，这一步也是为了知识模型生成语法树做准备。

Stanford CoreNLP工具是由斯坦福大学自然语言处理实验室研发的\cite{corenlp}，可以处理分句(ssplit)，分词(tokenize)，词性标注(POS)，词干化(lemma)，命名实体分析(NER)，语法树分析(parse)，短句情感分析(sentiment)和词语分析(natlog)等多种任务的工具集。本文中主要用到其中的分词，分句，词性标注，词干化，语法树分析等功能。

该工具集基于Java，由管道(Pipeline)将各子工具连接，从而方便地复用了各子工具。

以下代码为本文系统JAVA部分调用Stanford CoreNLP工具的代码：

new StanfordCoreNLP(

PropertiesUtils.asProperties("annotators",

"tokenize,ssplit,truecase,pos," +

"lemma,ner,parse,sentiment",

"tokenize.language", "en"));

图2-5 是Stanford CoreNLP 整体流程的说明，如图，文本被转化为Annotation对象然后输入各子工具集，各子工具集可选，但内部存在依赖顺序。

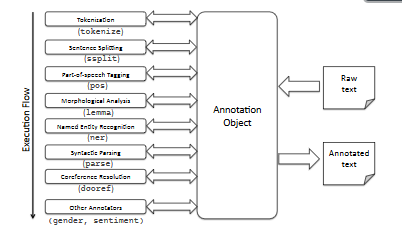


图2-5 Stanford CoreNLP整体流程图[51]

图2-6 则是Stanford CoreNLP2014年各子工具集的语言支持情况，可以看到，该工具集可以处理多种语言，其中对英文支持最为全面准确。同时，该工具集支持中文的句法分析，满足知识模型需求。

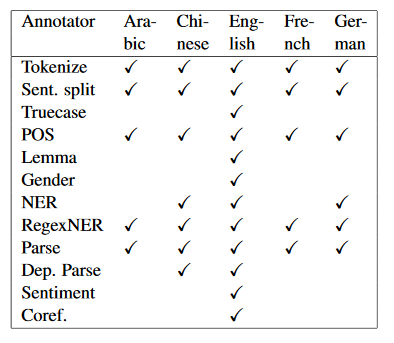


图2-6 Stanford CoreNLP语言支持情况[51]

**2.3.3 输入特征**

为了便于拓展模型和系统，本文将特征分为三类：

* 待编码的离散特征，为了便于表示，简称为Emb Feature。
* One-hot离散特征: Onehot Feature。
* 连续特征: C Feature。

三种特征各自有不同的处理方式。同时，对于某些离散特征，本文会删除词频较低的词汇，以提高模型拓展性，简称为 Del Feature，详细分析请见5.2.4小节。

**i.** **特征处理**

按照不同特征的类别，处理方式见图2-7，其中One-hot特征和Emb特征都需要转为编号，Emb特征需要进一步训练并转化为编码。

**ii.** **特征选择**

本文为不同模型及不同语言选取不同输入特征，如表2-4。

表2-4 各模型特征及类型

|  |  |  |
| --- | --- | --- |
|  | 中文 | 英文 |
| 知识模型 | 词汇(C) | 词性还原化词汇(C) |
| CNNPL模型 | 词汇编码(Emb,Del),  词性(Onehot) | 词汇编码(Emb,Del)，  词干编码(Emb,Del),词性(Onehot) |
| CNNLSTM模型 | 词汇编码(Emb,Del),  词性(Onehot) | 词汇编码(Emb,Del)，  词干编码(Emb,Del),词性(Onehot) |

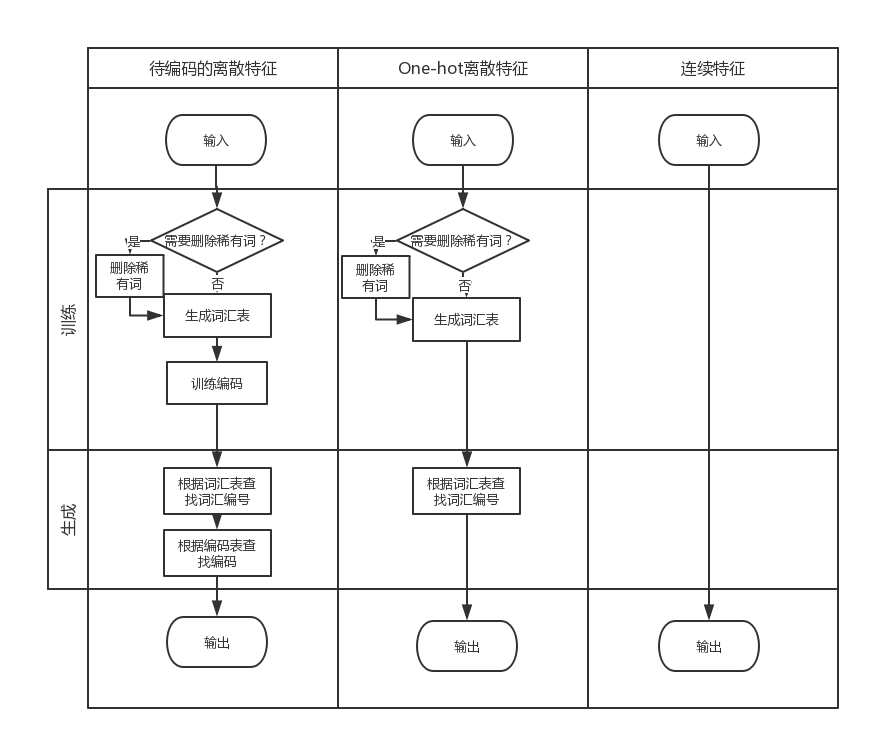


图2-7 特征处理基本流程图

**2.3.4 稀有词删除**

该步骤统计某特征在训练集中出现频率，之后将出现频率小于规定的最低频度的特征置换为一个特殊词"RAREWORD"。对于词性为nt（机构团体）, ns（地名）, nr（人名）, nz（其它专有名词）的单词所对应的特征，规定的最低频度通常大于全局最低频度，以删除专有名词，尽量减少这类无用词的影响。

本文中所用到的频率限制为：中文全局最低频率为7，专有名词最低频率限制为12。英文全局最低频率为12，由于英文文本的词性标注方式与中文文本不同，书名，机构名，人名等往往被识别为名词，因此很难识别英文文本中专有名词和普通名词的差别，故不设置特殊的专有名词最低频率。

**2.3.5 英文词性还原及词干化**

英文中动词往往具有多种时态，如过去式，现在进行时，一般现在时等，如got,gotten,get,getting,gets。同时，形容词和副词也常常具有同样的词干，如happy, happily。这些单词虽然形态不同，却具有语义上的联系，因此，有必要对单词进行词性还原和词干提取，以发现其内部隐含的语义联系。

本文使用Stanford CoreNLP对单词进行词性还原，之后再使用nltk Snowball Stemmer进行词干化，以此作为英文模型的输入特征。相关测试见附录2 词性还原和词根化工具准确率测试。

**2.3.6 特征生成详细流程**

本章详细介绍并举例说明模型的特征生成过程。其中，中文英文的生成流程略有差别。

**i.** **知识模型**

知识模型接受的输入为单词或者经过了词性还原的单词。当处理中文时，如图2-8，本文首先将进行编码转换工作，并将可能存在的繁体中文文本转化为简体中文，然后，本文使用ansj工具进行分词，将分词后的文本输入知识模型。而处理英文时，由于英文有较多时态变化，本文选择对英文单词做词性还原操作。如图2-9，本文首先进行编码转换，同时删除不应该出现在英文文本中的字符如汉语全角标点符号等，接着，本文使用Stanford CoreNLP工具进行分词，分词后的结果经过单词小写化处理后，还需要再使用Stanford CoreNLP进行词性还原，最后，词性还原后的单词才能输入知识模型进行分析。

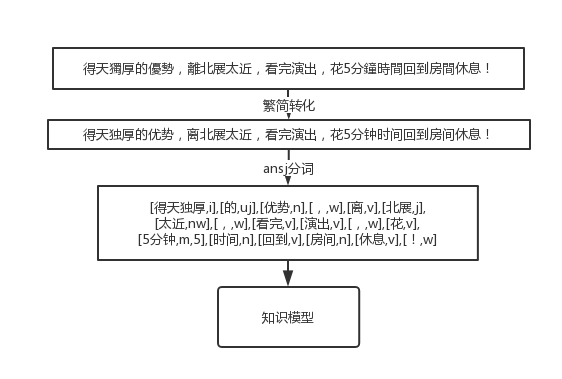


图2-8知识模型中文特征生成流程图

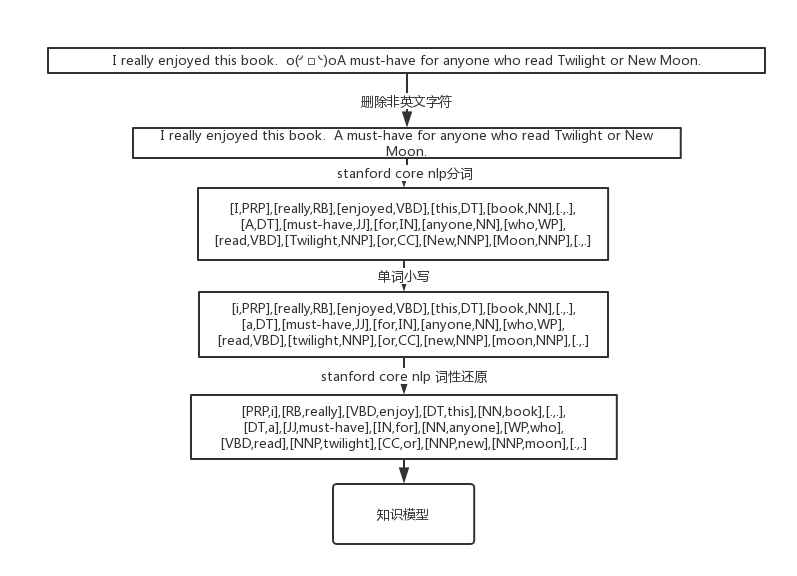


图2-9知识模型英文特征生成流程图

**ii.** **CNNPL模型**

神经网络模型接受的参数为文本转化成的固定长度的向量，其中CNNPL模型接受一组固定长度的向量，而CNNLSTMPL接受多组固定长度的向量直到序列被遍历完毕。本节将分别介绍CNNPL模型的中英文处理方式。

由于神经网络模型一次只能接受固定长度的向量作为参数，本文中CNNPL模型只取前半段文本中的词汇进行分析，定义CNNPL模型能处理的最大词汇数目为处理长度。如图2-10，在中文下，经过编码转换，分词工作后，本文删除词汇本身中的稀有词，并为词汇和词性编号，使用word2vec中的Continuous Skip-Gram模型\cite{mikolov2013b}将处理长度内的词汇转换为固定长度的向量，并将词性转化为onehot向量，最后，将二者拼接组成输入特征矩阵。

如图2-11,在英文下，经过删除非英文字符，小写转换，分词，词性还原工作后，本文使用nltk工具包中的snowball stemmer进行词根化，接着本文分别删除词汇本身和词根中的稀有词，为词汇，词根和词性编号，将处理长度内的词汇和词根都转换为固定长度的向量，最后将二者与词性onehot向量进行拼接。

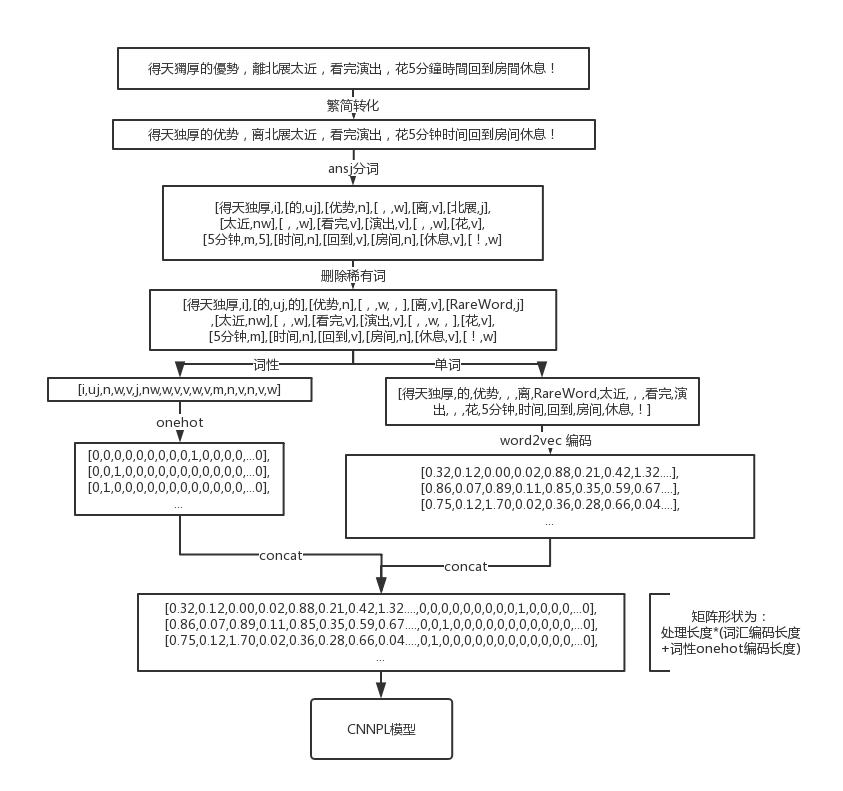


图2-10 CNNPL模型中文特征生成流程图

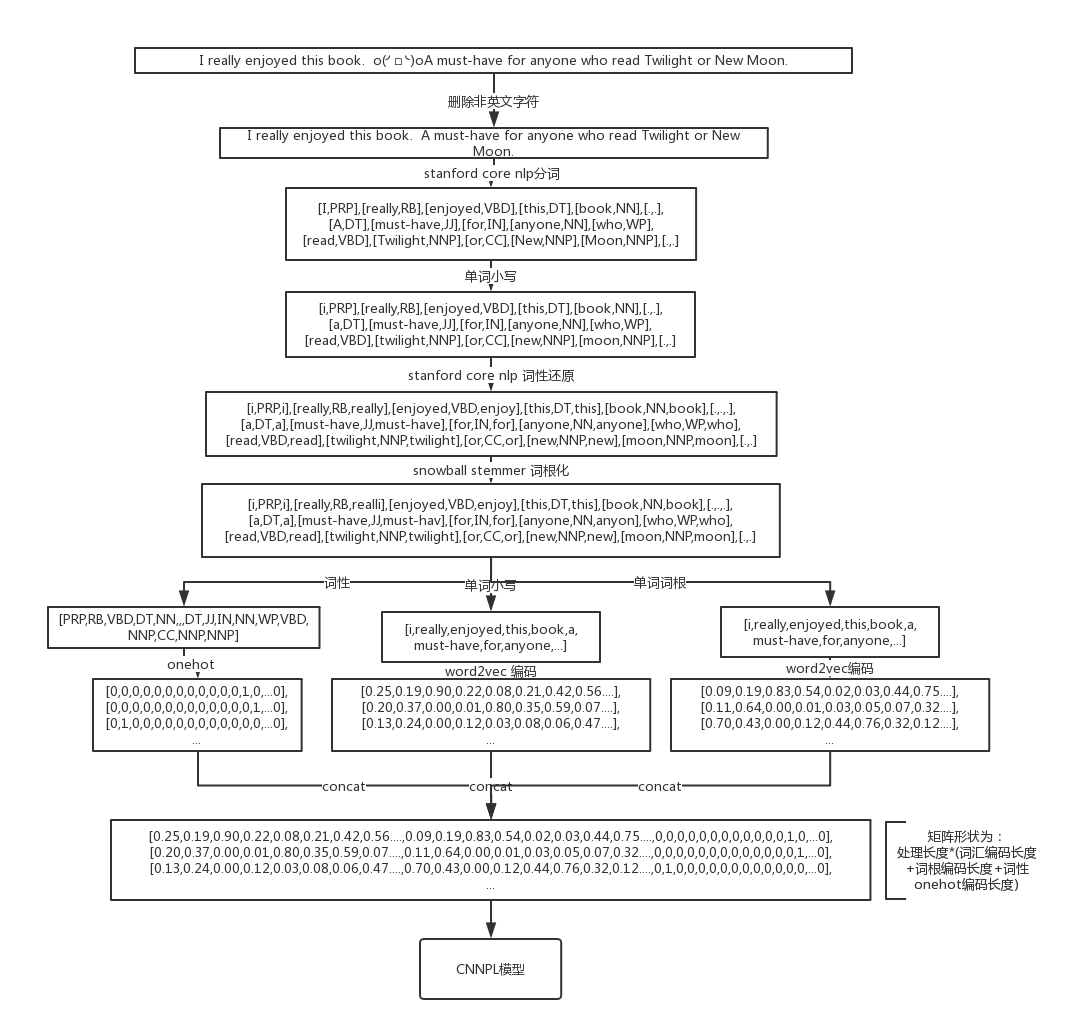


图2-11 CNNPL模型英文特征生成流程图

**iii.** **CNNLSTMPL模型**

作为循环神经网络，CNNLSTMPL模型需要进行反复迭代。本文将在迭代的某一步中，CNNLSTMPL能处理的最大词汇数目定为处理长度，将此时开始处理的第一个词的位置称为起点。如图2-12所对应序列，设处理长度为9，则一次生成的矩阵相当于：

[得天独厚,的,优势,，,离,北展,太近,，,看完],

[的,优势,，,离,北展,太近,，,看完,演出],

[优势,，,离,北展,太近,，,看完,演出,，],

[，,离,北展,太近,，,看完,演出,，,花],

......

CNNLSTMPL模型每次生成的矩阵中的每步的特征与CNNPL模型相同。为了使CNNLSTMPL模型能处理不定长序列，模型需要反复生成多次直到语句序列处理完毕。特征生成中文过程如图2-12所示，英文过程如图2-13所示。

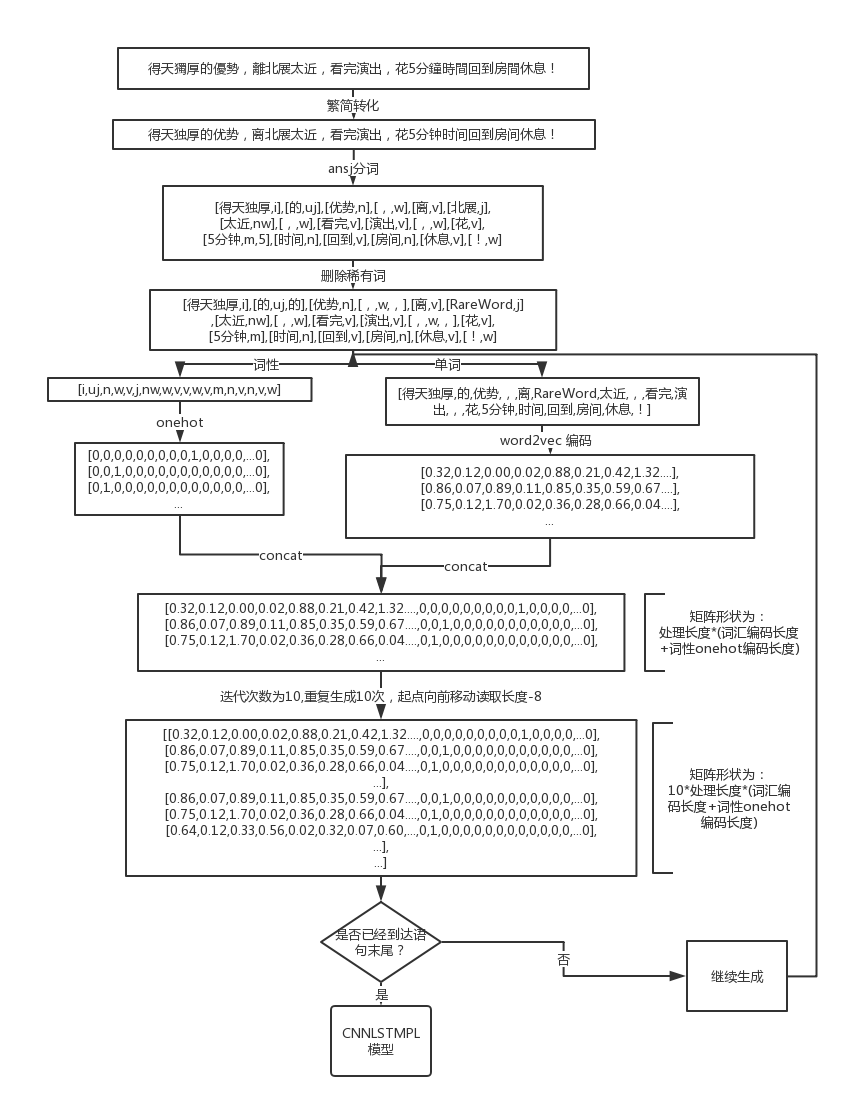


图2-12 CNNLSTMPL模型中文特征生成流程图

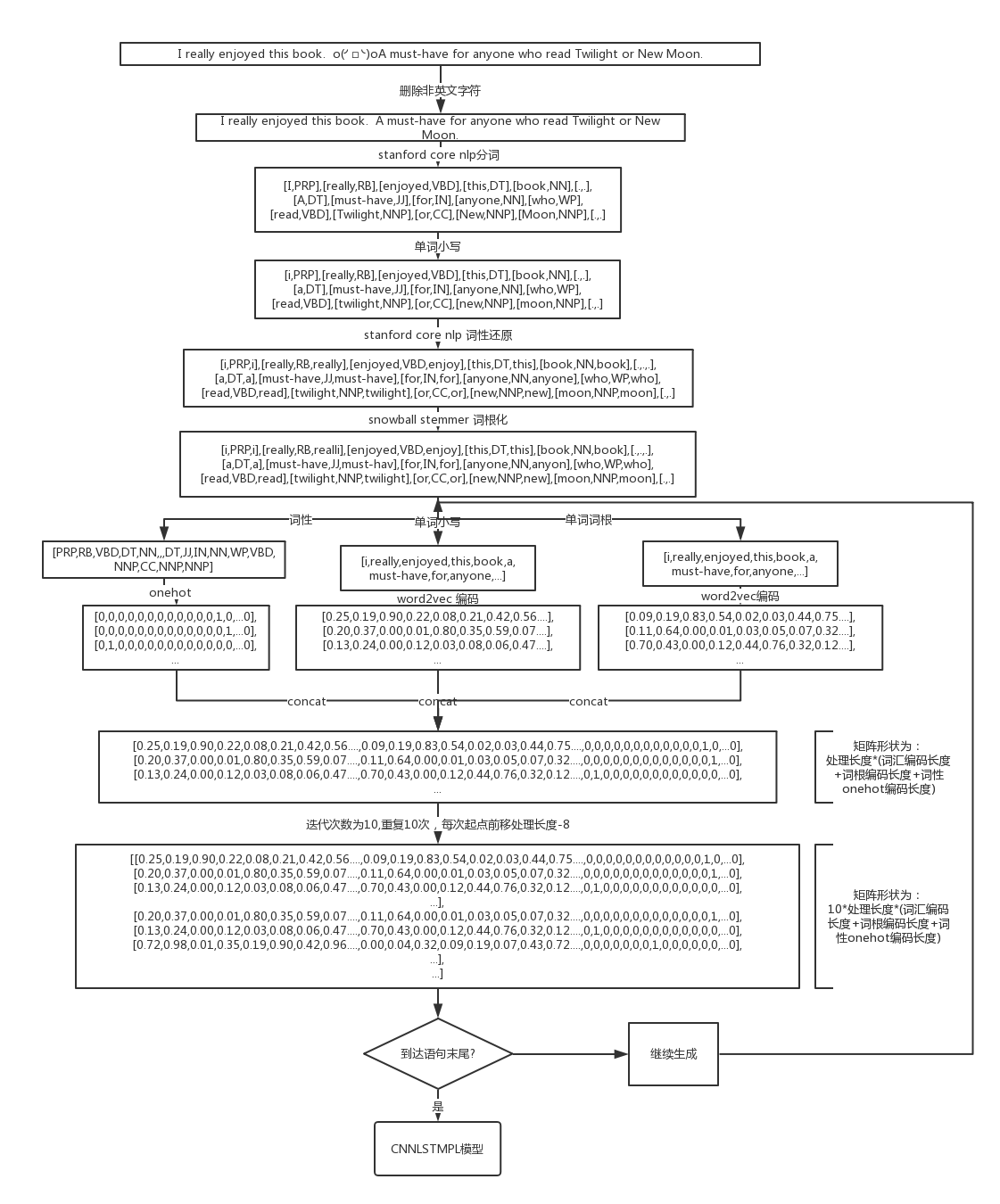


图2-13 CNNLSTMPL模型英文特征生成流程图

**第三章 基于知识的情感分析模型**

本章首先介绍了基于知识的情感分析模型所使用到的Stanford语法树，然后介绍了知识模型的主要思想和基本实现步骤。关于知识模型的性能分析详见第五章。

**3.1 知识模型简介**

基于情感词典的情感分类方法是基于知识的情感分类方法中的主流方法，本文所实现的知识模型也主要利用情感词典完成情感极性和情感强度的判定。但单纯通过情感词典判定单词或者短语的情感强度常常忽略说话者的隐含信息，如"她聪明但很令人讨厌"，其中聪明是正向形容词，"令人讨厌"是负向形容词，正向词与负向词的比例为1:1，恰好平衡，可能为矛盾情感，而实际上，因为"但是"起转折作用，"很"起强调作用且二者都与"令人讨厌"有关，该句的情感极性应该为负向。为了利用修饰关系或者语法特征，研究者们往往需要编写复杂的规则如[16]，这容易导致程序过于复杂，不易维护和拓展。基于语法树的方法则能通过利用语法树的层级关系判断句式和词语之间的依赖关系和作用范围，更加简便地处理情感传播，极性冲突和极性转换等问题[17]。

本文实现的知识模型主要使用Stanford 语法树和第2.2节所介绍的情感词典。该知识模型通过top-down方式，由语法树的根节点向叶节点递归，由词典或子节点推测每个节点的程度强度或情感强度。由于语法树的使用，知识模型往往可以避免引入冗长的规则，仅使用简单的规则处理复杂句式。例如，对于双重否定句"这家酒店不是不好。"，"好"的情感强度为+1.0，而"不"起否定作用，程度强度为-1.0，这会导致节点"不好"的情感为-1.0，而"不是"与"不好"是"不是不好"的子节点，而"不是"的程度强度也为-1.0，最后，"不是不好"的结果为+1.0，这与双重否定句所代表的实际语义相符。该模型不需要训练，适合处理情感较为强烈，使用情感词较多，语法较为规范的语句。

本文主要使用两个词典，一个为情感词典emotion，另一个为程度词典degree。emotion中的词条为情感词及对应强度，degree中的词条为程度词和对应强度。在词典中，强度可能为+2.0,+1.5,+1.0,+0.5,-0.5,-1.0,-1.5,-2.0。对于情感词来说，如果某情感词为正向词，则其对应强度为正数，反之，如果某情感词为负向，则其对应强度为负数。如"高兴"强度为1.0，"沉痛"强度为-2.0。对于强度词来说，如果某个词起强调作用，则强度为正数，起转折作用，则其强度为负数。例如"很"的强度为+2.0，"不"的强度为-1.0。

**3.2 Stanford 语法树**

Stanford Parser是由斯坦福自然语言处理小组实现的语法分析工具，也是Stanford CoreNLP的其中一个组件。它主要基于概率上下文无关文法（Probabilistic Context Free Grammar，PCFG，又称随机上下文无关文法，Stochastic Context Free Grammar，SCFG），能进行句法分析和语义依存分析（Semantic Dependency Parsing, SDP），生成语法树和依赖关系，是目前比较主流的一款语法分析工具。

本文选择Stanford Parser作为句法分析器，因此在此简单介绍Stanford Parser的中心算法思想。

定义：一个概率上下文无关文法是一个五元组，其中为非终结符集，为终结符集，为开始符集，为产生式集，对于任意产生式，其概率为。规则表示形式为：,其中为非终结符，为推导出的概率，即，该概率分布必须满足如下条件：[52]。

对应于语法树，相当于非叶节点，为叶节点，唯一且为根节点，为可行的路径，为路径转移概率。

在实践中，一般设为训练集语料中的最大似然估计，也即，其中为已标注语料中出现该规则的次数，为语料中出现该语法单元的次数。给定语法树后，就能计算出已标注词性的文本符合语法树的概率，而寻找对应的语法树可以转化为动态规划问题求解。

明显，PCFG是CFG(上下文无关算法)的拓展，但上下文算法不含概率，无法解决语言的二义性问题，因此PCFG要比CFG更适合于自然语言处理。

但由于PCFG等价地对待词组中的所有词，导致相同结构的PCFG对词汇信息和结构偏向不明显。词汇化的PCFG文法为每个语法树的每个节点添加首要词(HEAD)，从而克服了PCFG文法的弱点，取得了巨大的成功[53]。

本文使用Factor模型，在Stanford Parser Factor模型中词汇化的PCFG方法用于同时表示语法结构与依赖关系，如图3-1(c)。Klien等[54]人认为，设为词汇化的PCFG树，则可以被一棵语法PCFG树 (如图3-1(a))和一棵结构相符合的依赖关系树 (如图3-1(b))确定。于是，基于文本构建的问题就被转化为基于文本构建一对，，使最大，这里代表文本同时符合，的概率。Klein等人进一步假设，相对独立，也即，并分别使用动态规划构建概率最大的，，之后使用A\*启发式算法搜索和的近似结合方式，该算法取得了不错的成果。

Stanford Parser还对分析器做了一系列优化[55] [56] [57] [58] [59] [60] [61]，在此不做赘述。

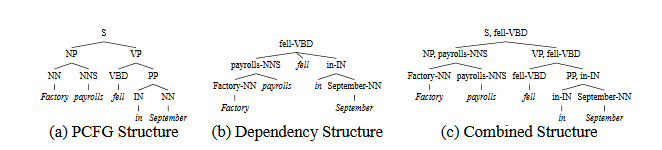


图3-1 三种分析结构

**3.3 基本流程**

算法基本流程如图3-2：

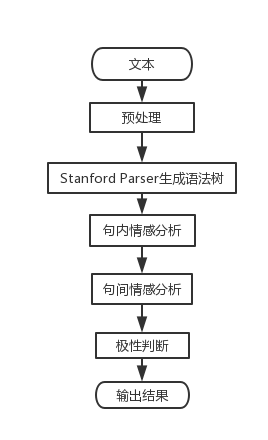


图3-2知识模型基本流程图

伪代码如下：

def analyze(words, lang):

if lang == 'zh':

trees = parser\_zh.parse(words)

else:

trees = parser\_en.parse(words)

ans = 0

for tree in trees:

dfs(tree.root,tree)

ans += tree.root.sentiment

if ans == 0:

if tree has child whose sentiment != 0:

return contradict

else:

return neural

else if ans > 0:

return positive

else:

return negative

def dfs(node, tree):

node.phrase = ''

if not isleaf(node):

for child in node.daughterTrees:

dfs(child, tree)

if isleaf(node):

node.phrase = node.label

else:

for child in node.daughterTrees:

if lang == 'zh' or isW(child) or child.label == "n't":

node.phrase = node.phrase + child.phrase

else:

node.phrase = node.phrase + ' ' + child.phrase

found, node.sentiment = lookup\_in\_emotion(node.phrase)

if not found:

found, node.sentiment = lookup\_in\_degree(node.phrase)

if not found:

if not isleaf(node):

d = e = 0

for child in daughterTrees:

if child.label in ['RB', 'VE', 'AD']:

d += child.sentiment

else:

e += child.sentiment

if d == 0:

node.sentiment = e

elif e == 0:

node.sentiment = d

else:

node.sentiment = d \* e

算法首先将分词后的词汇输入到Stanford Parser中，每个短句将生成一棵独立的语法树。

接着算法对每个短句所生成的语法树采用top-down递归遍历。

在递归的过程中，算法会首先在词典中查找子树所对应的短语，如果找到了对应的短语，则将短语所代表的情感强度或者程度强度设为该子树的强度，否则，分别统计子树的子节点的情感强度和程度强度，取其乘积作为该子树的强度。

最后，算法将多棵语法树根节点的情感强度相加，得到整体情感强度。如果整体情感强度大于0，则最终情感极性为正向，如果整体情感小于0，则最终情感极性为负向，如果整体情感强度等于0且子树情感强度都为0，则为中性，如果整体情感强度等于0，但存在子树情感强度不为0，则为矛盾情感。

如图3-3，对于“This is the right size for any use! Uses little space and looks extremely good on the counter.”，Stanford Parser将会生成图中所示结果。

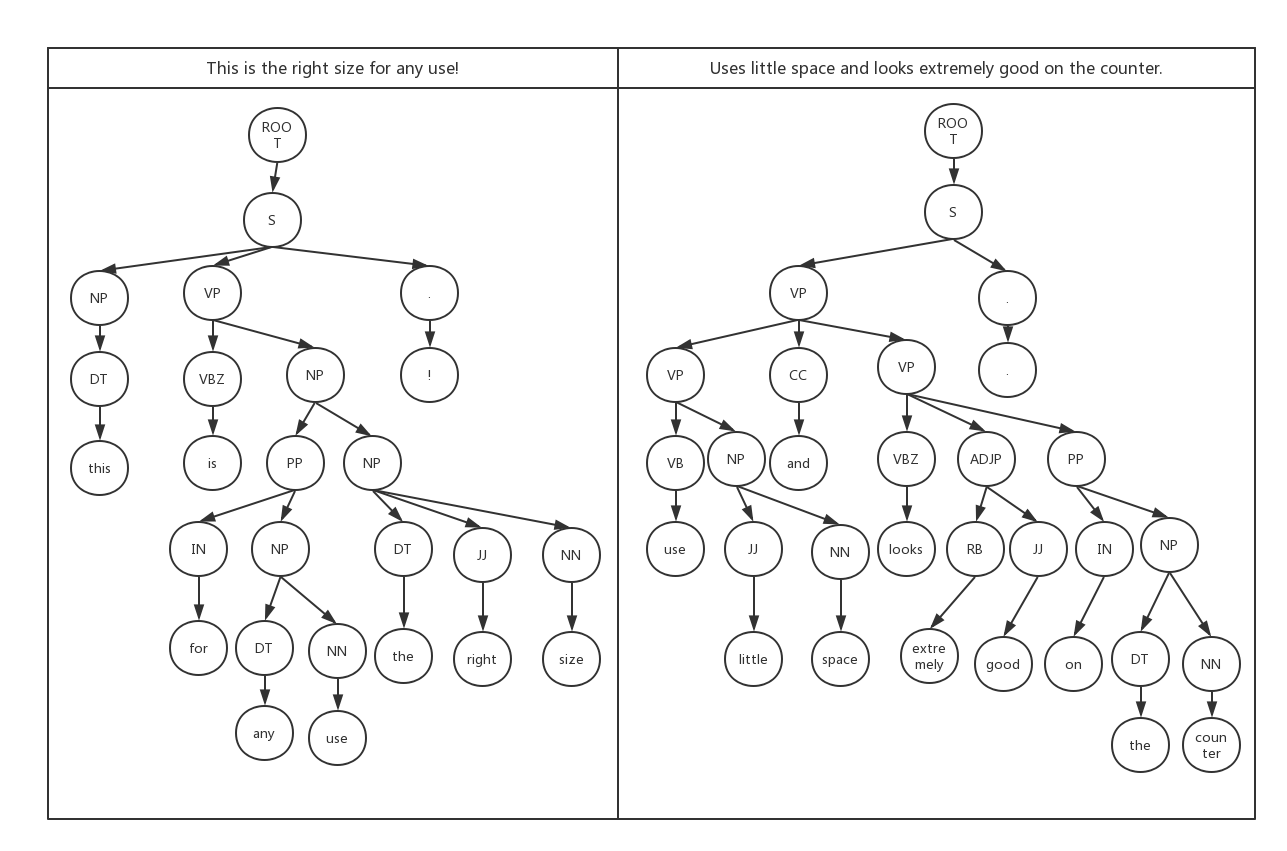


图3-3语法树可视化

算法分别从左右两棵树根节点开始分析，逐步到达叶节点，以右边的树为例，该树对应"Uses little space and looks extremely good on the counter."。

1. 在情感词典中，"little"的情感强度为-1.0，因此"little space"，"Use little space"的情感强度都被标注为-1.0。
2. "extremely"在程度词典中被标注为+2.0，"good"在情感词典中被标注为+1.0，因此"extremely good", "looks extremely good on the counter."被标注为+2.0
3. 综上，"Uses little space and looks extremely good on the counter."被标注为+1.0
4. 右边的树也遵循此流程，极性为+1.0，将子句合并考虑，得到该语句整体情感强度为+2.0。
5. 由于算法将大于0的结果定为正向，所以该评论文本的整体情感为正向。

最终，算法可视化结果如图3-4。

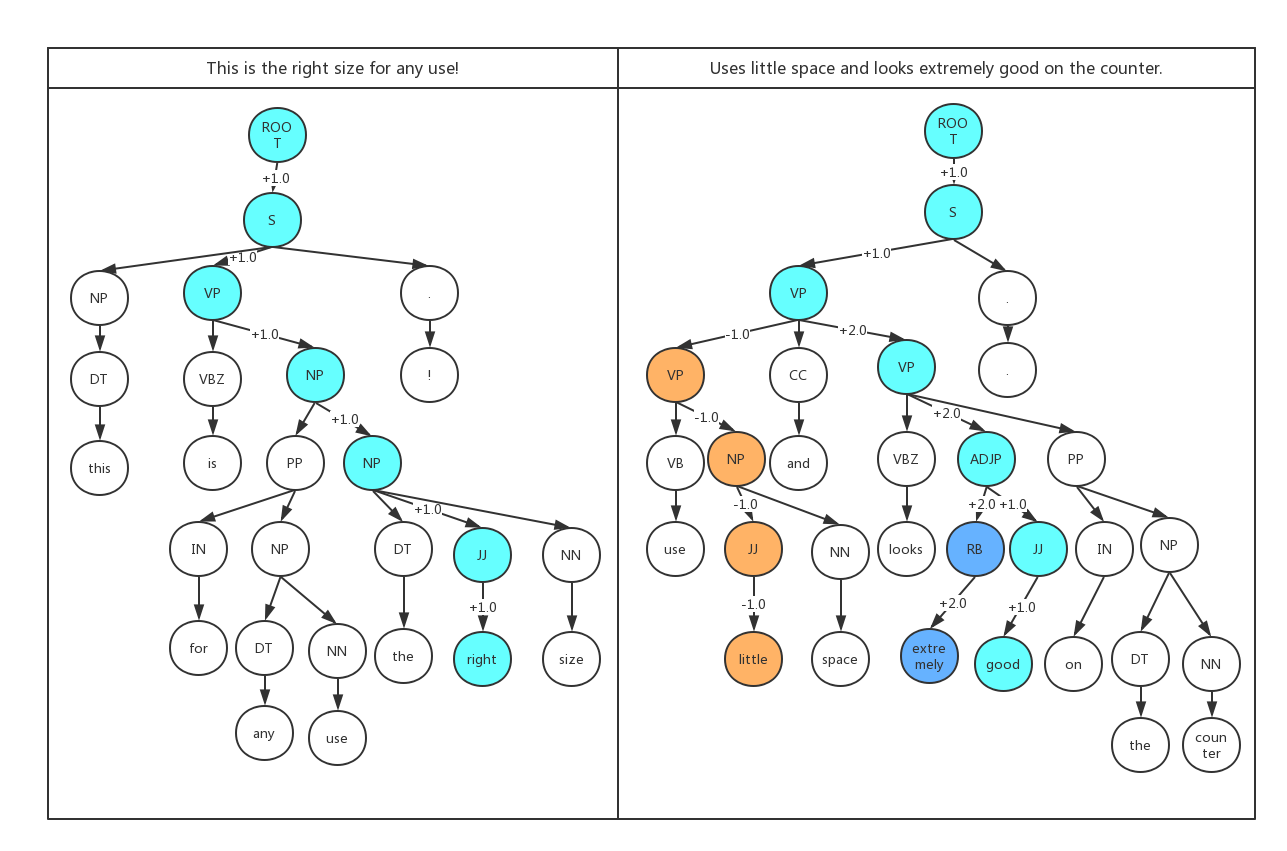


图3-4知识模型算法结果可视化

**第四章 基于神经网络的情感分析模型**

本章首先介绍了模型实现中用到的算法和相关知识，接着，本文分别介绍了由C&W模型改进的短语级卷积模型(Convolutional Neural Network on Phrase Level, CNNPL)和结合CNNPL与LSTM模型的短语级带有LSTM细胞的卷积模型(Convolutional Neural Network with LSTM on Phrase Level, CNNLSTMPL)。CNNPL和CNNLSTMPL的性能分析详见第五章。

**4.1 深度学习相关算法**

本章主要介绍三部分，一、基础的人工神经网络(Artificial Neural Network, ANN),二、有关卷积神经网络(Convolutional Neural Network,CNN)的知识，三、有关循环神经网络的知识。其中ANN部分介绍ANN的概念，反向传播(Back Propagation, BP)网络，常用激活函数，本文所用到的损失计算函数，常用的优化函数。CNN部分介绍卷积，池化的概念，同时介绍C&W模型。RNN部分介绍RNN，同时介绍长短时记忆型循环神经网络(Long Short Term Memory, LSTM)。各网络之间的关系如图4-1。

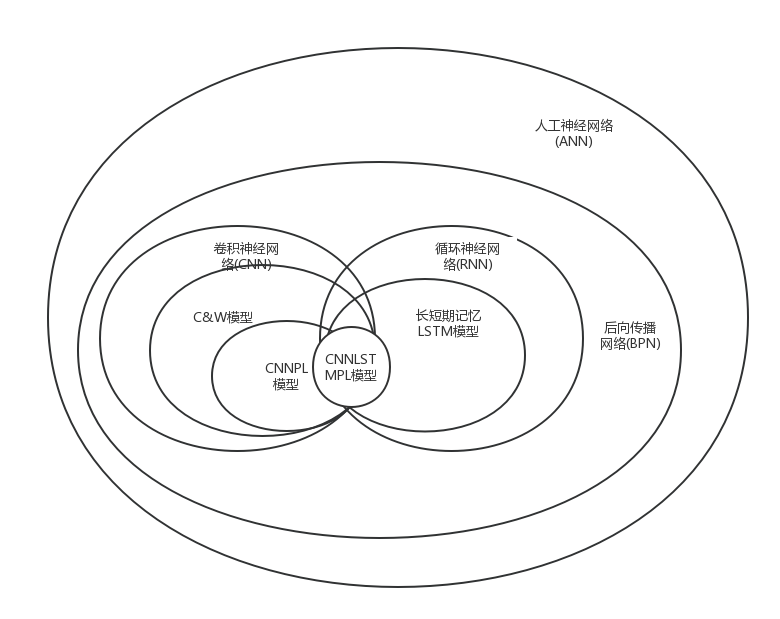


图4-1 神经网络关系图

**4.1.1 人工神经网络**

人工神经网络提供了一种普遍且实用的方法从样例中学习值为实数，离散值或向量的表示方法。ANN在一定程度上受到生物大脑中神经元相互连接的模型的启发，它由许多基本单元构成，每个单元有一定的实值输入（该输入可能来自外部也可能是其它单元的输出），同时产生单一的实数值输出。常见ANN的基本单元如图4-2:

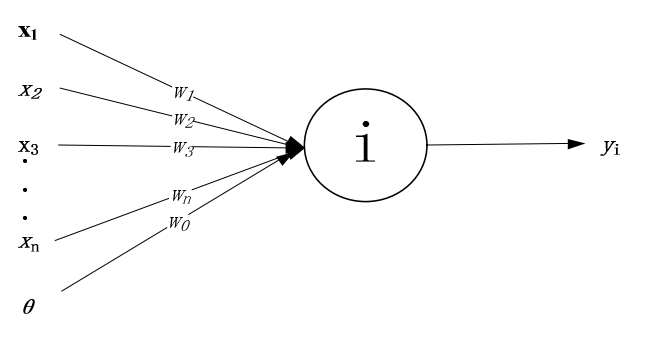


图4-2 ANN基本单元[62]

其函数可以写作：

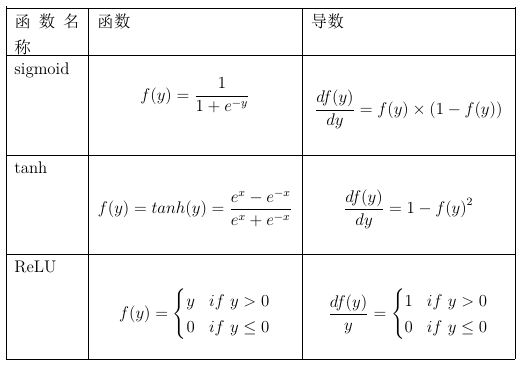


其中为输入，为激活函数，负责将线性输出转变为非线性输出以增加神经网络的表达能力，为权重(weights)，为偏置(bias)，与具有相同形状。与通常是神经网络训练的主要对象。

**i.激活函数**

常用的激活函数有sigmoid(又称logistic函数),tanh,ReLu(Rectified Linear Unit)等，具体公式如下：

表4-1 激活函数公式及导数



由于sigmoid函数和tan函数的导数是线性输出的二次多项式，在训练时易导致梯度爆炸或梯度消失问题，故本文中线性层之间的激活函数主要选用ReLU函数。

一个较典型的人工神经网络通常由多层基本单元构成，如图4-3，通常有输入层，隐含层，输出层，各层之间通过激活函数将线性输出转化为非线性输出。

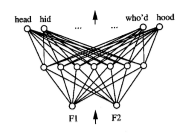


图4-3典型ANN结构[62]

**ii.** **损失计算函数**

本文中主要使用带有softmax的交叉熵函数计算网络输出值和目标值之间的误差。softmax函数是sigmoid函数在多分类问题上的拓展。



其中为网络的输出，其长度为类别总数。



其中为网络的目标值，其长度为类别总数。

**4.1.2 反向传播网络**

反向传播算法是目前多层神经网络主要的训练方法，它主要基于微积分的链式求导法则(如式(4-4))，对神经网络从输出层向上进行快速求导。



反向传播神经网络的训练基本流程是：

1. 初始化网络。
2. 首先将输入沿前向传播，求出误差。
3. 使误差沿网络反向传播，计算梯度，从而使用优化方法更新权值。
4. 如果达到终止条件，跳出循环，否则，重复第2-4步。

**4.1.3 优化方法**

在神经网络上训练时，其假设空间为所有可能的实数权向量的集合（将偏置等参数也视为权向量），一个高维空间。当误差函数和目标值已知时，假设空间中会形成一个误差曲面，而训练的目的就是尽可能找到该误差曲面的最小值。

如图4-4是二维假设空间中误差曲面的假想图，可以看出该误差曲面是具有单一全局最小值的抛物面，沿梯度方向可以到达该最小值点。但实际问题中误差曲面往往具有更高的维度和更为复杂的形状，因此使用恰当的下降方法即优化方法是非常重要的。

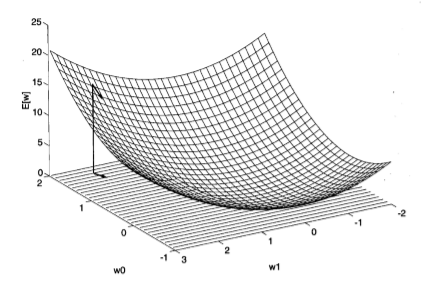


图4-4误差曲面[62]

**i. SGD方法**

为了确定一个使误差最小化的权向量，梯度下降搜索（Gradient Descent, SGD）从初始权向量出发，每一步都沿梯度方向修改该权向量，直到到达全局最小误差点。

该训练法则可以写为[62]：



其中为误差相对于分量方向上的梯度，为控制步长的学习速率。

但梯度下降方法每一步都需要计算所有训练样例上的整体误差，同时，梯度下降方法很可能收敛到局部极小值。

为了缓解这些困难，人们提出了小批量随机梯度下降算法（Mini-batch Gradient Descent），该算法根据每个训练样例batch单独计算误差来更新权值，相当于为每个batch单独定义不同的误差函数。因此，如果误差平面上有多个局部极小值，随机梯度下降算法可以通过不同的误差避免陷入局部最小值。目前梯度下降搜索的实现方式基本为小批量随机梯度下降算法，因此，本文中提到的SGD算法代指小批量随机梯度下降算法（Mini-batch Gradient Descent）。

**ii. ADADELTA算法**

本节主要介绍目前流行的优化方法ADADELTA算法，为了介绍ADADELTA算法，本节首先介绍ADADELTA算法改进的对象-Adagrad算法。

**ii.1 Adagrad算法** 虽然随机梯度下降方法为避免陷入局部最小值提供了可行的方法，但梯度下降方法对所有需要更新的权向量都使用了全局学习速率，而全局学习速率可能不适应于所有的参数。Adagrad(Adaptive Subgradient)方法[63]能够对每个参数自适应不同的学习速率，对稀疏特征使用更大的学习速率，因此更适应于处理稀疏数据。

其学习速率为：



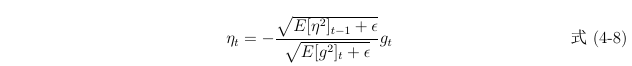
其中,为t时刻对于权重的学习速率，为初始学习速率，为平滑常数，用于防止分母为0，代表j时刻方向上的梯度。

**ii.2 ADADELTA算法** Zeiler等[64]人认为Adagrad方法存在三个问题:

1. 其学习率单调递减，训练后期学习率过小。
2. 需要手工设置一个全局的初始学习率。
3. 更新$X\_t$时存在单位不统一现象。

因此Zeiler等人提出了ADADELTA方法，用于改进Adagrad方法。ADADELTA算法基于牛顿迭代法，其学习速率不再基于全部梯度平方之和，而主要基于最近的梯度。

其学习速率更新规则为：





其中为t时刻的梯度，为衰减速率，为平滑常数。

**iii. ADAM算法**

ADAM(Adaptive Moment Estimation)[65]根据损失函数对每个参数的梯度的一阶矩估计和二阶矩估计动态调整针对于每个参数的学习速率，其中一阶矩的作用类似于冲量项(momentum)，二阶矩则与ADADELTA算法相似。ADAM算法迭代更新步长有一个较为稳定的范围，因此更适合RNN。







实践中取=0.9，=0.999，=1e-8。

**4.1.4 卷积神经网络**

卷积神经网络(Convolutional Neural Network, CNN)是一种前馈网络，该网络内部单元的连接模式不同于传统的全连接方式，每个神经元的输入仅与上一层对应单元核大小范围内的输出单元有关。卷积神经网络相当于在神经网络上执行卷积操作。

数学上的卷积函数定义如下：

设和为在上的可导函数，则称为函数与的卷积，记作

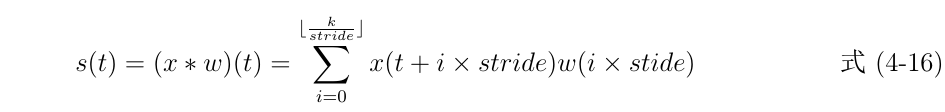
其中被称为输入，被称为核函数(kernel function)，在统计学上，卷积操作相当于加权平均。

由于自然语言为离散序列，故本文使用的卷积运算核为1维，卷积公式可以写为：



在实际使用中，a的取值范围通常较小，以减少运算，产生局部输出单元的综合输出。

下采样(downsampling)卷积函数能够每隔一段距离采样，称该距离为步长(stride)，则当核为1维，核长为k时，卷积公式可以写为：



卷积操作通过三个重要的思想帮助改进机器学习系统，即稀疏交互，参数共享和平移不变性[66]。首先是稀疏交互，如图4-5，传统神经网络单元会与上一层的每个输入单元产生交互，而通过限制卷积核大小，单元只会与附近的输入单元产生交互，从而大大减少了需要训练和存储的参数，提升了效率和泛化能力。然后是参数共享，在卷积神经网络中，多个位置共享同样的参数集合，而不需要针对每个位置单独训练。最后是平移不变性。如果我们在输入中移动一个事件，那么对应的输出仍会输出而且会延后相应长度。例如，对于“手机 小巧 玲珑”和“我 觉得 手机 小巧 玲珑”，当卷积核的大小为2，步长为2时，两句中的“手机 小巧”将会输出同样的结果。

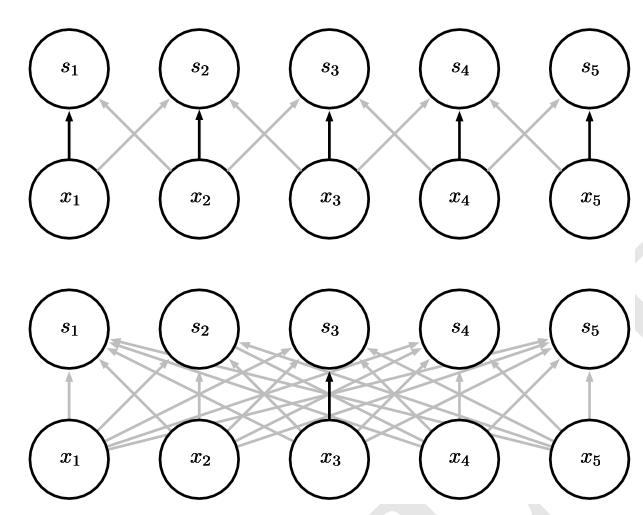


图4-5 CNN与ANN对比[66]

**i. 池化**

池化(Pooling)是卷积神经网络中非常重要的操作，能帮助学习输入特征的不变性，也能减少特征数量，增加计算速度。常见的池化方式有两种，最大池化和平均池化。最大池化给出了卷积层输出中一个矩形区域内特征的最大值，而平均池化则为矩形区域内特征的平均值。池化相当于简单使用某一位置附近区域的总体输出来代替各单元的输出。在自然语言处理中，最大池化的作用往往要好于平均池化，因此，本文采用最大池化方式。

**4.1.5 C&W模型**

C&W模型是由collobert等于2011年[36]提出的基于卷积神经网络的自然语言处理框架，该模型被设计能够适用于几乎所有自然语言处理任务，同时也能对某个任务进行细化以提高精度。C&W模型分为两种模式，单词窗口和语句模式，后者主要结构如图4-6。模型可以分为七层，输入层，查表层，卷积层，池化层，线性层和对应的激活函数层，最后是输出层。C&W模型在各任务上都表现良好，接近为特定任务特化的系统性能，因此本文选择C&W模型作为基本框架。

**4.1.6 循环神经网络**

循环神经网络(Recurrent Neural Network)是专门用于处理序列数据的神经网络。如图，传统神经网络为序列中的每个状态分别训练权重，但RNN在各时间步内共享参数。给定时间t的变量后，t+1时变量的条件概率分布是平稳的，不依赖于t，也即RNN具有时间上的平移不变性。

对于长度为t的序列，一般神经网络计算方式可以表示为：



其中代表所有可训练的网络参数。

而RNN的计算方式可以表示为:



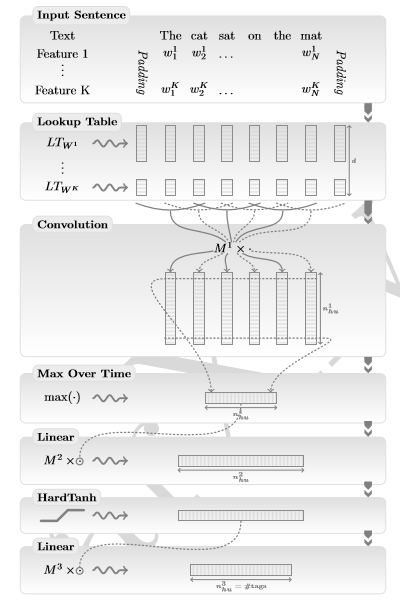


图4-6 C&W语句模式基本模型[36]

RNN能将长度为t的序列映射为固定长度的向量，但这种映射通常是有损的，因此，需要保证足够丰富，能够记住序列中的信息。

RNN主要有三种结构，图4-7中每个时间步都有输出，并且隐藏单元与过去隐藏单元相关联，图4-8中隐藏单元与过去输出相关联，而图4-9中只有最后一个时间步有输出。由于语句级情感分析任务最终只有一个输出，所以本章中RNN结构符合图4-9。

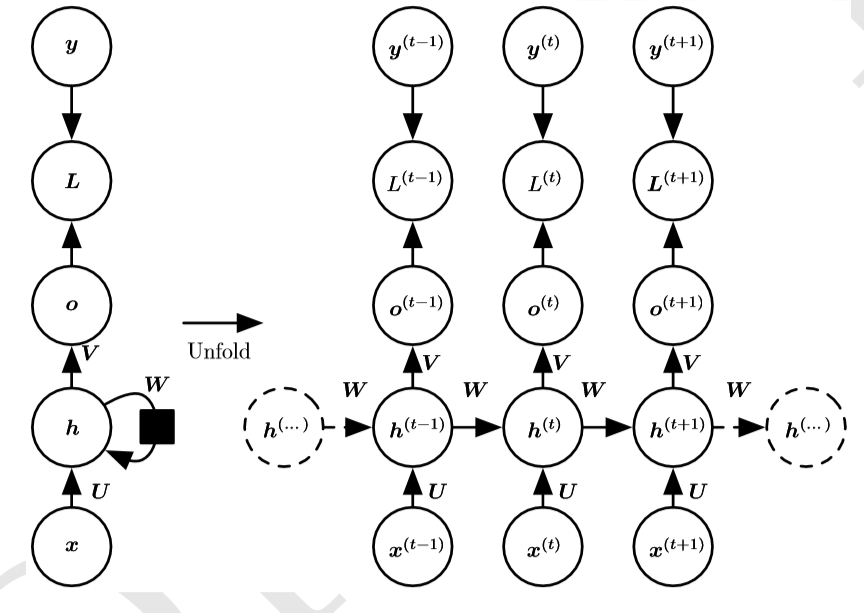


图4-7 RNN基本结构a[66]

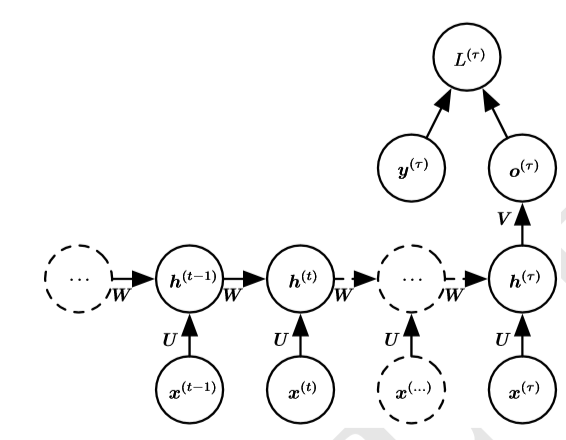


图4-8 RNN基本结构b[66]

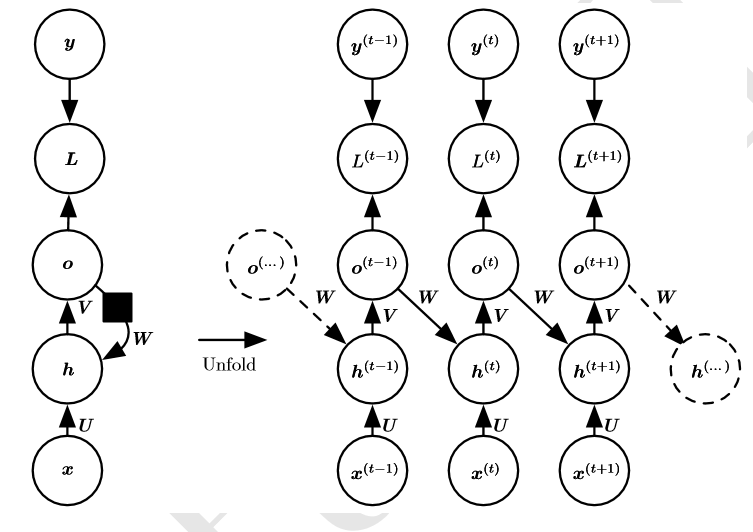


图4-9 RNN基本结构c[66]

训练中，RNN使用通过时间反向传播算法(BPTT)反向推导t步梯度。实践中，为了降低复杂度，同时也因为梯度爆炸和梯度消失的限制，t往往取较小的定值。

**4.1.7 LSTM**

RNN为信息的持久化提供了行之有效的方法，但RNN存在梯度爆炸或梯度消失的问题。

考虑最简单的RNN单元，即：



该公式相当于：



若W符合下列形式的特征分解：



则原式可以转化为：



因此，随时间成指数形式增长，若幅值小于1时，将随t的增长趋向于0，出现梯度消失现象，而当幅值大于1时，又激增，导致梯度爆炸

为了减轻梯度消失问题，Hochreiter等[67]提出了LSTM模型，该模型主要由以下四部分构成：

1. 遗忘门：控制状态更新
2. 输入门：控制信息保存。
3. 输出门：控制读取信息。
4. 记忆单元：存储状态。

LSTM的具体计算流程如下：



其中为输入门输出的结果，为输入门的参数，为上一时刻的输出，为本时刻的输入，为输入门的参数偏置。



其中为记忆单元得到的结果，为记忆单元的参数，为输入门的参数偏置。



其中为遗忘门输出的结果，为遗忘门的参数，为遗忘门的参数偏置。



其中为本时间点的状态。



其中为输出门输出的结果，为输出门的参数，为输出门的参数偏置。



其中为最终输出结果。

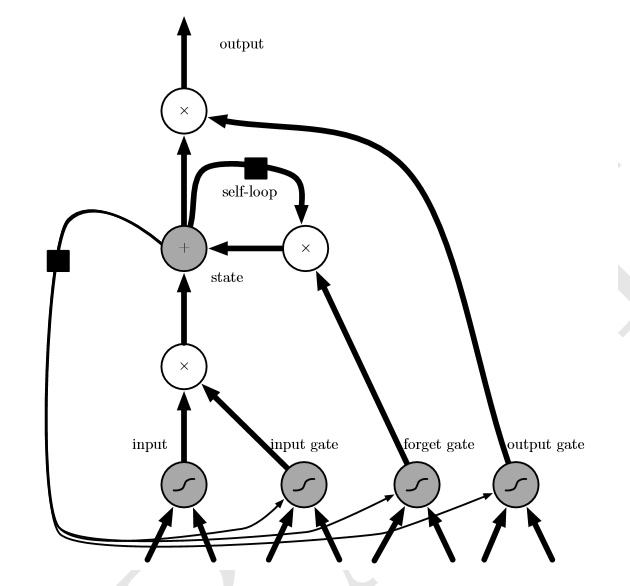


图4-10 LSTM基本结构[66]

**4.2 CNNPL模型**

作为一个处理自然语言任务的框架，C&W模型能够通过训练查表层将单词转化为向量，同时，它还能通过卷积神经网络抽取局部短语特征，最后，C&W模型能够通过线性层结合局部短语特征将句子转化为固定长度的向量，从而完成文本分类。但在实际应用中，局部短语特征所保留的语法结构往往会对情感分类任务造成不必要的干扰，因此，本文针对C&W模型在情感分类任务上的应用，提出了由C&W模型改进的，抛弃了全局语法特征的短语级卷积模型(Convolutional Neural Network on Phrase Level, CNNPL)。相比于C&W模型，该模型有以下改进。

1. 选择全局池化以提取全局短语特征，而不是C&W模型中的卷积时延模型，从而舍弃语法特征，提高泛化能力。
2. 将特征分为两类：待编码Emb特征与Onehot特征。待编码特征通过word2vec中的Continuous Skip-Gram模型进行训练，将训练后的向量作为查表层的初始化向量以加快训练速度。Onehot特征则直接将离散特征转化为固定长度的Onehot编码向量，避免对Onehot编码进行训练。
3. 直接截断语句，取前半部分计算，以加快计算速度，减少空间损耗。
4. 舍弃头部PADDING，若句长小于处理长度，则尾部直接补零，以减少时间损耗。

**4.2.1 神经网络结构**

如图4-11，CNNPL模型主要分为八层，输入层，卷积层，全局最大池化层，Dropout层，三层线性层和输出层。

其中，输入层对应于C&W模型的输入层和查表层，输入的离散特征在输入层转换为Onehot编码或者经过训练的编码，连接层将输入的特征包括连续特征进行连接，拓展每个词所对应的特征向量维数。输入层设计为与具体特征选择无关，以便于调整模型，也便于未来用不同的特征对模型进行拓展。

经过反复试验，本文中卷积层由一维五个卷积核构成，本文中取[3,5,6,7,9]，不同的卷积核分别对应不同的池化层。本文中选取的池化层都为全局池化层，经过全局池化层可以提取出文本在不同长度上的全局特征。令人惊讶的是，同样经过反复试验，全局池化层的效果好于局部池化层或者堆叠局部池化层，笔者认为局部池化可以保留语句结构，而全局池化可以去除语句结构信息，而在神经网络对语句分析的过程中，过于丰富的语句结构可能反而起了抑制泛化的作用，详细参数分析见附录3 卷积核测试。

Dropout层被加在卷积层之后以便于提高模型泛化性。Dropout能够随机使一些卷积层的输出失效，以迫使模型利用其它输出结果，从而尽量减少特征与某一特定输出之间的关联。

模型接下来使用了三层特征数目依次递减的全连接层即Linear1，Linear2，Linear3抽取高级特征，线性层之间使用ReLU函数进行激活，以增强表达能力，提取高级特征。

最后模型进行误差和精度计算，并使用ADAM优化函数进行反向传播。

由于该模型相对于特征独立，所以该模型可以被同时应用于中英双语，也可以拓展到其它语句分类任务上。

**4.2.2 模型主要流程**

模型分为训练模式与测试模式：

在训练模式下，基本步骤为：

* 1. 数据生成器按不同的语言要求生成输入特征，对超过限定长度的语句直接截断，以语句前半部分生成输入特征，实践证明往往前半部分已经拥有足够判断情感极性的信息。
  2. 模型将这些特征通过输入层和连接层转化为统一的向量，计算误差。
  3. 通过优化函数Optimizer，将误差反向传播以训练参数。
  4. 每隔特定步数，对验证集进行测试，测试时不启动反向传播。
  5. 若验证集精度大于已有最好精度，保存该模型，同时在测试集上进行测试，本文认为在测试集上的精度衡量了其泛化能力。

在测试模式下，基本步骤为：

* 1. 载入模型。
  2. 使用数据生成器将单一语句转化为输入特征。
  3. 计算各标签的概率，即Output层中的Logits，输出概率最高的标签。

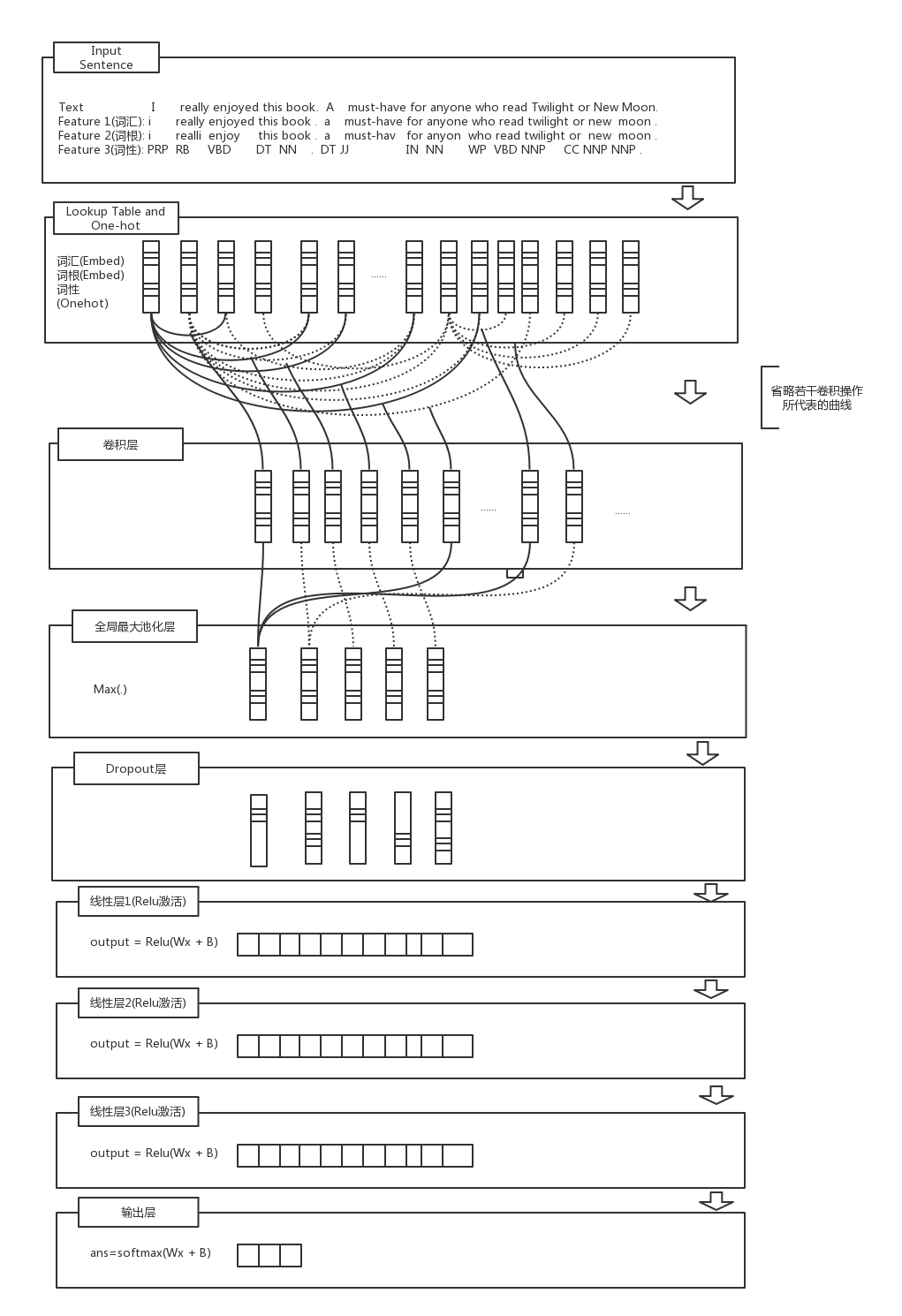


图4-11 CNNPL模型整体结构

**4.3 CNNLSTMPL模型**

虽然CNNPL模型在数据集上表现良好，但该模型无法处理变长语句，也不能利用自然语言的序列性质，因此，本文结合CNNPL模型与LSTM模型，提出了短语级的带有LSTM细胞的卷积模型(Convolutional Neural Network with LSTM on Phrase Level)。CNNLSTMPL模型结合了循环神经网络与卷积神经网络的特点，能够在提取全局短语特征的同时处理变长序列。相较于CNNPL模型，它能利用自然语言的序列属性训练更好但更有效的参数。实践中，CNNLSTMPL模型往往能比CNNPL模型取得更好的精度成果。

**4.3.1 神经网络结构**

如图4-11，CNNLSTMPL模型主要分为八层，输入层，卷积层，全局最大池化层，Dropout层，第一线性层Linear1，LSTM单元层，第二线性层Linear2和输出层。其中输入层，卷积层，池化层，输出层与CNNPL模型相同。因为RNN能够有效地利用基于时序的局部特征的特性，采用与CNNPL模型相同的全局最大池化层可能违反直觉，但实际上不使用卷积神经网络或者使用局部池化以及堆叠局部池化反而导致模型后期精度下降。本文认为这是由于语法结构反而会导致过拟合。在实际测试中，该网络结构的性能较为出色，在NLPCC2014SCDL数据集的测试中超过了竞赛中第一名的成绩。

在对输出特征进行dropout后，输出首先进入Linear1而不是LSTM层，以减少特征数量，抽取高级特征，便于LSTM层进行保存。最后，模型通过Linear2以及输出层这两层线性层计算输出结果。换而言之，LSTM层取代了CNNPL模型中的第二线性层。

**4.3.2 模型主要流程**

模型分为训练模式与测试模式，设stepNum为LSTM单元迭代次数，在训练模式下，基本步骤为：

1. 数据生成器按不同的语言要求生成stepNum步输入特征。
2. 模型将stepNum步输入特征全部通过输入层转化为矩阵，进行余下的计算并计算误差。
3. 执行第1-3步直到该batch中的句子全部遍历完毕。
4. 通过优化函数，将误差反向传播以训练参数。
5. 每隔特定步数，对验证集进行测试，测试时不启动反向传播。
6. 若验证集精度大于已有最好精度，保存该模型，同时在测试集上进行测试，本文认为在测试集上的精度衡量了其泛化能力。

在测试模式下，基本步骤为：

1. 载入模型。
2. 使用数据生成器将单一语句转化为stepNum步输入特征。
3. 计算各标签的概率，输出概率最高的标签。
4. 重复执行第2-3步直到该句子被遍历完毕，输出最后一次迭代中概率最高的标签。

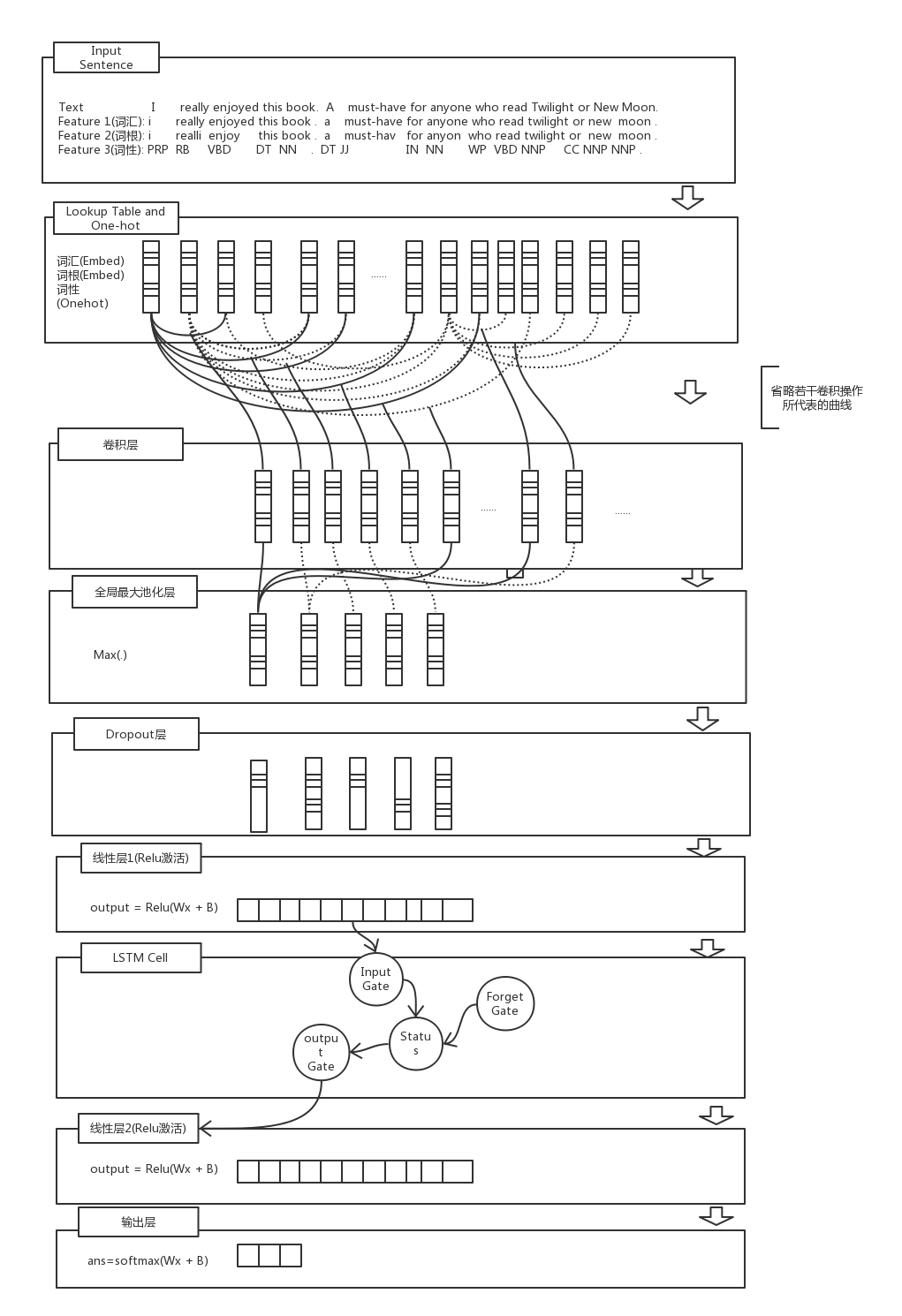


图4-12 CNNLSTMPL模型整体结构

**第五章 结果分析**

本章主要对三个模型的准确率和时空复杂度进行测评，此外，本章对神经网络模型中的各项参数进行了分析。

**5.1 模型分析**

**5.1.1 实验条件**

为了证实文本提出的知识模型，CNNPL模型和CNNLSTMPL模型的有效性，本章使用NLPCC 2014的SCDL数据集和用于展示的携程数据集进行测评，实验条件如下：

运行参数：

1. 操作系统: ubuntu16.10 64位
2. 运行平台: tensorflow v1.1
3. 磁盘空间: 1.9 TB
4. 内存空间: 32 GB
5. 处理器: Intel Core i7-6700 3.40 GHz x 8

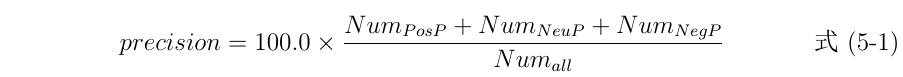
神经网络参数:

1. CNNPL的处理长度定为99\%词数，CNNLSTMPL的处理长度定为99\%长度的四分之一。
2. CNNPL和CNNLSTMPL模型的初始学习率都为0.001。
3. 离散待编码特征(Emb特征)的编码长度统一定为32。
4. dropout层的保存概率定为0.6。
5. CNNPL中第一线性层的输出特征维数为256，第二线性层的输出特征维数为128，第三线性层的输出特征维数为64。
6. CNNLSTMPL中第一线性层的输出特征维数为256，LSTM的状态维度为256，第二线性层的输出特征维数为64。
7. CNNLSTMPL模型的迭代次数为10。
8. 每个batch中包含200条语句，每处理40个batch后验证模型并决定是否保存模型。

**5.1.2 精度分析**

本文中验证标准为准确率，即分类正确的语句占测试集的百分比。

具体计算方法为：



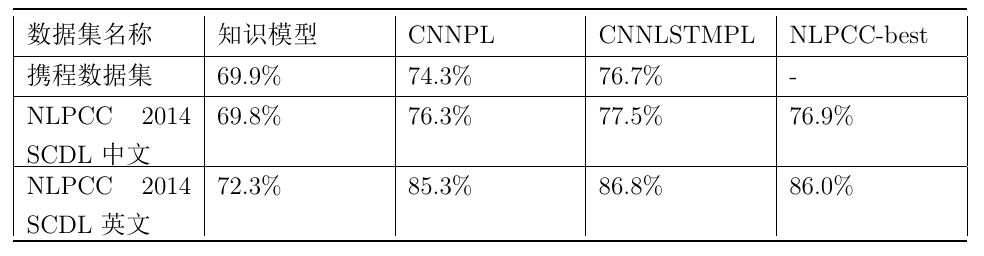
其中为测试集中预测值与目标值都为正向的文本数目，为测试集中预测值与目标值都为中性的文本数目，为测试集中预测值与目标值都为负向的文本数目，为测试集中全部的文本数目，

神经网络模型的测试方法为：

1. NLPCC 2014的SCDL数据集在官方训练集上进行封闭训练，使用官方测试集作为测试集。
2. 携程数据集使用5-fold进行测试，依次选取其中一个子集作为验证集，一个子集作为测试集，最终的测试准确率为五次平均。

准确度结果如表5-1：

表5-1 准确度结果



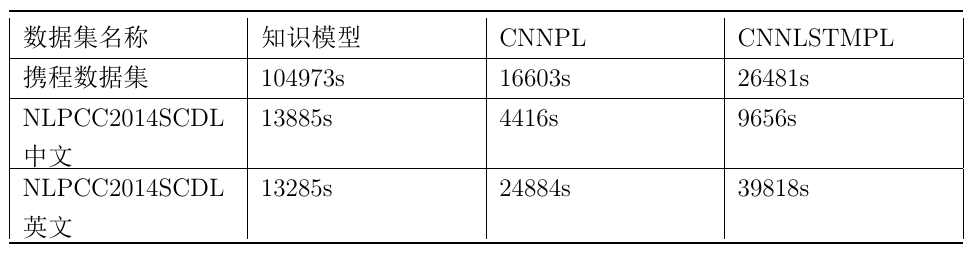
其中NLPCC-best为官方正式比赛中最高的准确率[68]，这里由于NLPCC测试集中正向:负向为1:1，所以该准确率为，其中PP为官方公布的正向准确率，NP为负向准确率。

从表中可以得出，CNNLSTMPL和CNNPL在数据集上的表现都较好，其中CNNPL接近于第一名的准确度，而CNNLSTMPL略高于第一名的准确度，知识模型的表现则略差。由此可以证明CNNLSTMPL和CNNPL模型是有效的，而知识模型还需要再加以改进。同时，CNNLSTMPL模型的精度往往高于CNNPL模型，这可能是由于CNNLSTMPL模型一次处理的单词较少，从而需要训练的参数更少，参数更有效，而且CNNLSTMPL模型更能利用自然语言的序列信息。综合来看，基于深度学习的神经网络不需要大量语言学知识，也不依赖于情感词典和语法分析工具，却可以通过学习特征表达的方式达到超过传统的基于知识的模型的性能。

**5.1.3 时空复杂度分析**

时间消耗如表5-2（取达到最高精度的时间）：

表5-2 时间消耗结果



由于CNNLSTMPL模型需要反复迭代，因此CNNLSTMPL模型速度较LSTM模型更差，而且由于训练策略的原因，CNNLSTMPL模型并不特意将长度相似的语句放在同一个batch中，这导致短句处理完毕后模型需要处理长句导致的补零，如果长句均匀分布在训练集内，就会导致训练速度下降。

知识模型速度较慢的原因可能有四点：

1. 知识模型启动时需要同时启动ansj和stanfordParser的中英版本，耗时较多。

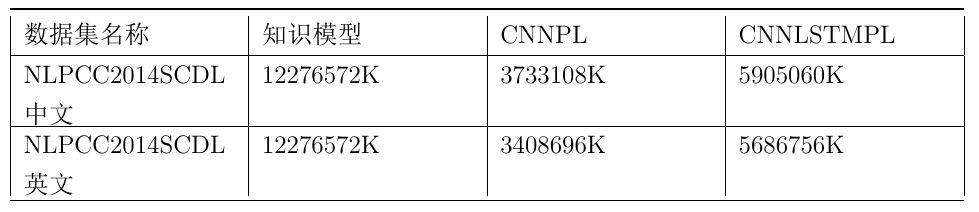
2. 知识模型在实现中不直接使用分词后的词语，而是主动调用分词工具，因此导致时间较慢。

3. 知识模型需要对每条语句建立语法树，因此耗时较多。

4. 知识模型通过jpype调用java实现，中间操作较多，且jpype需要管理和启动java虚拟机。

空间消耗如表5-3：

表5-3 空间消耗结果



从表中可以总结出在三种模型中CNNPL的速度最快，空间消耗率是最低的，因此，可以得出结论，在文本较短情感较为强烈时，使用CNNPL模型是节约时间和空间的做法，而如果需要处理不定长的文本，或者对精度要求较高，则应当使用CNNLSTMPL模型。

**5.2 参数分析**

本节分析对比神经网络模型的各种基本参数以及策略，由于神经网络训练较为费时，所以本节选取CNNPL模型，使用NLPCC2014SCDL中文数据集进行训练，以对比不同参数或策略条件下的精度，得出最优参数或策略。

**5.2.1 初始编码策略**

在预处理过程中，本文使用word2vec训练待编码的离散特征向量。为了探究该步骤是否存在意义，本文使用CNNPL模型，在参数相同的情况下分别运行了三次，第一次使用word2vec初始化编码，第二次使用零矩阵初始化编码，第三次则使用截断的正态分布随机初始化向量，取mean=0,stddev=0.1，准确率随训练过程的变化如图5-1：

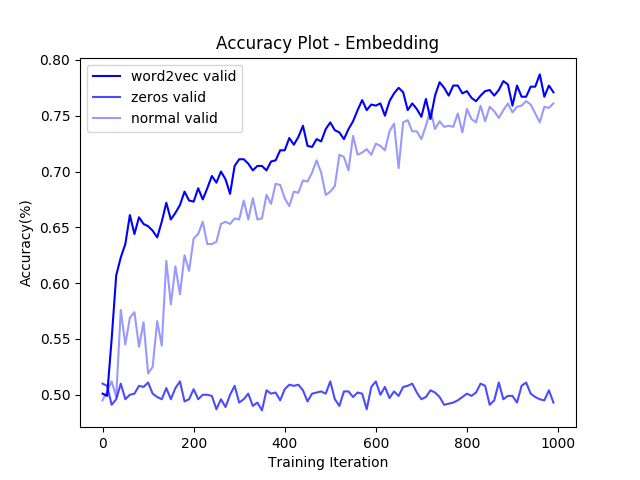


图5-1 编码-准确率变化图

通过观察该图内的三条曲线，可以发现word2vec的编码准确率较高，正态分布的编码准确率次之，但也呈现上升趋势，并且随时间逐渐接近word2vec编码的准确率，而零矩阵则无法训练，几乎保持为一条直线。综上所述，可以认为word2vec编码加快了训练速度，并且可能提高模型准确率。因此，使用word2vec初始化编码是有意义的。

**5.2.2 优化函数选择**

本文中CNNPL模型和CNNLSTMPL模型都使用ADAM算法作为优化函数。本小节中对本文中使用的优化函数做出评估，分别使用SGD算法，ADADELTA算法和ADAM算法在同样的参数下各自训练CNNPL模型，得到准确率随时间的变化曲线如图5-2:

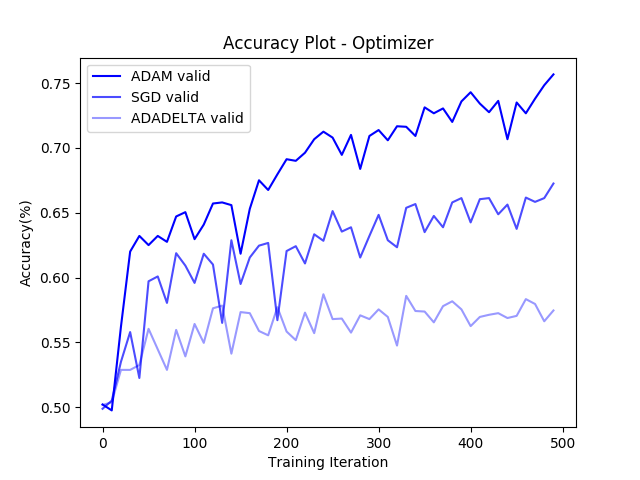


图5-2 优化函数-准确率变化图

通过观察图中的三条曲线，可以发现ADAM的训练速度和准确率都最高，SGD次之，而ADADELTA最慢。实际上，这是因为ADADELTA算法陷入了局部最小值点。综上所述，可以认为选择ADAM作为优化函数是正确的。实际上，SGD算法也是学术研究中较为常用的算法，如果人工配置的学习率足够强大，往往能够得到自适应梯度的优化函数所不能得到的效果。

**5.2.3 激活函数选择**

本文中模型都使用ReLU算法作为线性层之间的激活函数。本小节中对本文中使用的激活函数做出评估，分别使用ReLU算法，tanh算法和sigmoid算法在同样的参数下各自作为线性层之间的激活函数训练CNNPL模型，得到准确率随时间的变化曲线如图5-3:

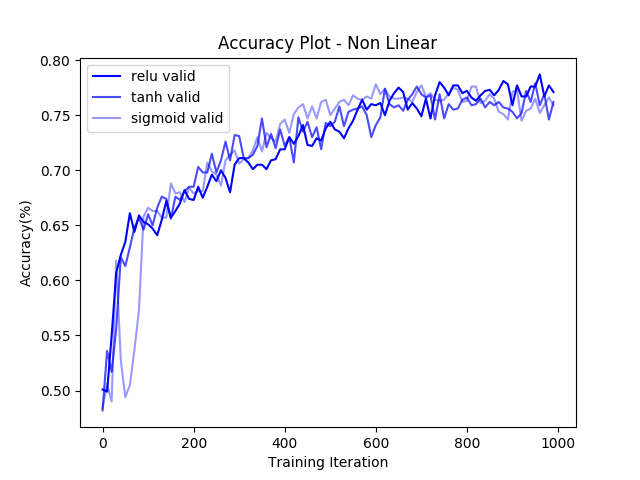


图5-3 激活函数-准确率变化图

在训练过程中，三种函数所对应的模型都保持近似于一致的上升趋势，但当训练进入后期，接近最高准确率时，ReLU也即颜色最深的曲线的准确率保持最高，因此，可以认为ReLU相较于sigmoid函数和tanh函数来说更适宜作为线性层之间的激活函数。这可能与ReLU具有单侧抑制和稀疏激活性有关。

**5.2.4 是否过滤稀有词**

本文在预处理过程中会对离散特征做过滤稀有词操作，即按照词性和最低频度将频度小于此词性的最低频度的离散特征转化为一个特殊的词，"RAREWORD"。本小节探讨该操作的必要性。

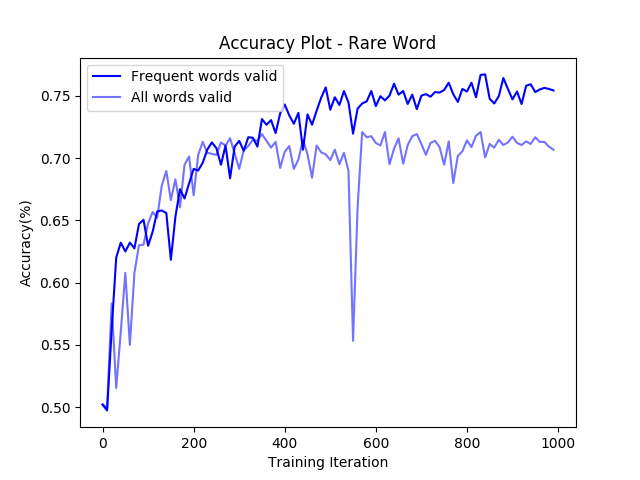


图5-4 是否过滤稀有词-准确率变化图

如图5-4，明显，使用全部单词，不进行稀有词过滤的模型准确率较低，泛化性能较差。从另一方面来说，例如数据集中只有一条负向文本中出现了"北京饭店"这个稀有词汇，则神经网络极有可能将"北京饭店"学习为负向情感词汇，但实际上该词汇是中性的实体词汇。通过过滤稀有词，模型能够尽力保证词汇在多种句式中出现，从而能够学习到词汇的真正含义，避免被标注影响，也能帮助抽取高级特征。

**5.2.5 是否使用dropout**

dropout函数随机使输出单元的一部分为0，即使其输出无效。使用dropout函数往往能够迫使模型学习更加健壮的特征，避免被局部特征所误导，可以减轻局部最小值的问题。但dropout本身可能导致模型精度无法提升等问题。本小节讨论神经网络模型在处理情感分类任务中dropout是否必要。如图5-5，分别进行了两次训练，一次dropout 保持不变的可能性设置为0.6，另一次不设置dropout。

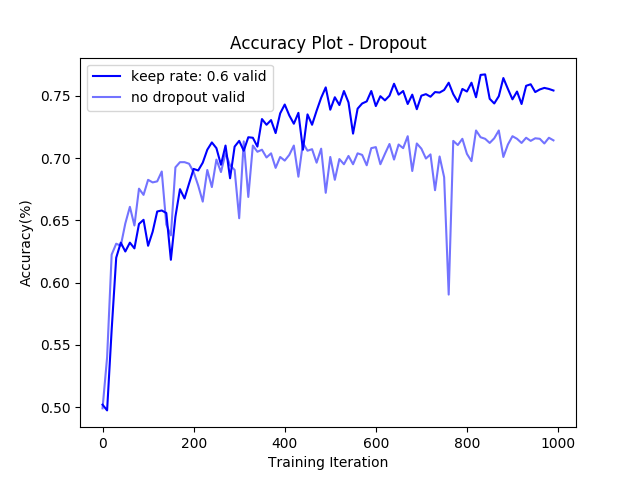


图5-5 是否使用dropout-准确率变化图

从图中可以看出，尽管在训练的早期阶段无dropout的模型准确率上升速度更加快，但在后期使用dropout的算法能明显增加准确度。因此，可以认为使用dropout不仅能影响到准确率，而且还是必要的，能够帮助泛化。在实践中，为了体现dropout的效果，本文将之设为0.6。

**5.2.6 是否使用词性**

本文将词性作为输入特征之一输入神经网络，本小节讨论神经网络模型在处理情感分类任务中词性(Part-of-Speech, POS)是否必要。如图5-6，本文分别进行了两次训练，一次使用词性和去稀有词的单词(Word + POS)作为输入特征训练模型，另一次不使用词性，仅使用单词(Word)作为输入特征训练。

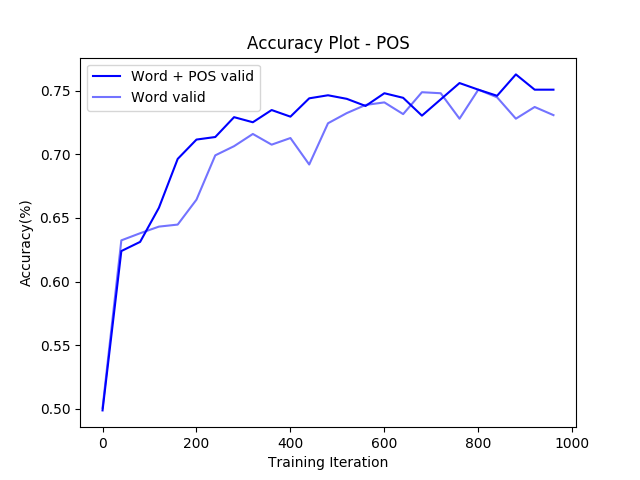


图5-6 是否使用词性-准确率变化图

从图中可以看出，使用词性进行训练时，在相同的迭代次数下，准确率上升更快，同时，使用词性进行训练时的最大准确率比不使用词性训练更高。可以认为，这是由于词性填补了删除特殊词造成的语义空白，为"RareWord"添加了补充信息。

**5.2.7 word2vec编码训练方式**

本文使用待训练的数据集进行word2vec编码训练以初始化查表层。而目前的主流方法往往是使用一个足够丰富的语料库训练word2vec编码。本小节讨论使用何种策略训练word2vec编码结果更优。结果如图5-7，本文分别进行了两次训练，第一次使用NLPCC2014SCDL中文数据集作为语料库训练，第二次将NLPCC2014SCDL中文数据集和携程语料库合并作为语料库训练初始编码。

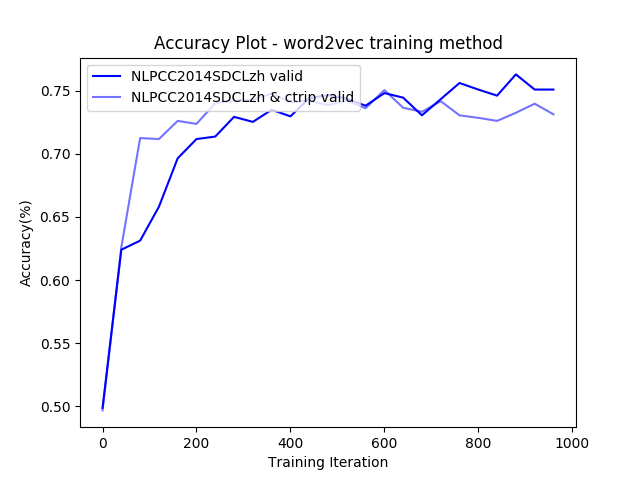


图5-7 word2vec编码训练方式-准确率变化图

从图中可以看出，添加携程语料库进行训练后，前期训练速度较快，但后期精度比只是用NLPCC2014SCDL中文数据集时更低。这可能是由于携程数据集是酒店领域的语料，而NLPCC2014SCDL中文数据集则是由书籍，CD，手机等领域组成的语料。由于二者所属的领域不同，在不同的领域中词汇表达的情感也不尽相似。如"地面太光滑"和"手机屏幕太光滑"，同样的词汇("光滑")很可能带有不同的隐含情感。因此，为了将训练语料局限在数据集相关的领域内，本文在没有附加的语料库的情况下选择使用训练集来进行word2vec编码训练。但本文不排除使用同领域的更为丰富的语料库训练初始编码将会给准确率或训练速度带来提高的假设。

**第六章 情感分析系统**

为了更好地展示模型效果，本文实现了一套实时判断评论情感的展示系统。用户在前端输入语句后，该语句会通过网络传入系统服务器后台，进行分析，最后返回前端由网页展示。

**6.1 整体模块**

如图6-1，系统主要分为两大模块即情感分析器和展示系统。情感分析器主要负责训练模型，预测输入文本的情感极性。展示系统包括服务器，前端网页和对应设计的RestAPI服务。整个系统由配置管理模块管理，以保证只需要改变参数就可以更改展示系统和模型参数，同时，配置管理模块也负责保存每次训练的模型参数，以便于记录和比较。

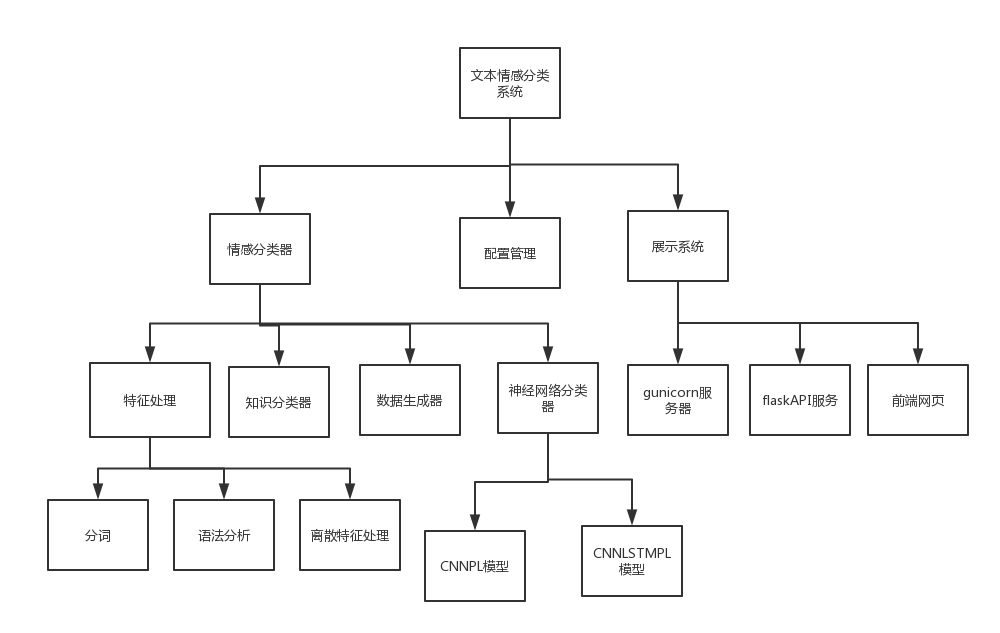


图6-1 系统模块图

**6.2 开发环境**

1. 编程语言: Java, python
2. 开发环境: Eclipse, Pycharm
3. 依赖管理: pip, maven
4. 版本管理: git
5. 数据库: mongoDB
6. 架构: Restful API架构
7. Server: gunicorn
8. Restful API: flask
9. 前端: 静态HTML + AJAX + JQuery + HTML5 + Bootstrap
10. 分词工具: Ansj, Stanford Parser
11. 英语词根化工具: nltk.stem.SnowballStemmer
12. 语法分析工具: Stanford Parser
13. 机器学习平台: tensorflow v1.1
14. java调用工具: jpype

**第七章 结束语**

本文研究如何使用深度神经网络对文本进行情感分类，实现了基于卷积神经网络的CNNPL模型，和结合卷积神经网络与LSTM的CNNLSTMPL模型，二者都在NLPCC2014SCDL数据集上取得了接近第一名的结果。与此同时，本文实现了基于知识和情感词典的知识模型，用于与神经网络模型相比较。

本文实现了一套可以同时应用于中英双语的情感分类器，在实现上，该分类器通过面向对象编程的思想，能够通过只加载配置生成保存不同的模型，便于研究对比。同时，本文基于Restful API实现了一套用于展示的网页。

项目从2016年10月末开始准备，一直到2017年6月初结束，前期主要实现语义级的情感分析任务，完成模型并调参，后期则致力于实体级的情感分类任务，但考虑问题不够完善，配套的两个模型没有达到足以写入论文的精度。

项目还可以改进和继续努力的地方有：

1. 寻找足够稳定，精度足够，结构有拓展性的局部卷积和池化方案。

2. 知识模型应考虑更加细致的规则，以提高精度。同时，应当考虑更加详细的情感分级，或者基于不同主题赋予单词不同的情感权重。

**参考文献**

空白第一页

空白第二页

空白第三页

空白第四页

空白第五页

空白第六页

**致 谢**

本毕设的完成得到了很多人的帮助。首先要感谢吴国仕老师的指导与帮助，老师的认真与负责使我受益良多，毕设后期没有听从您不要涉足实体领域的劝告，结果果然惨淡首尾。同时也要感谢李晶老师，在跟随李晶老师和吴国仕老师做大学生创新竞赛时，我对这一领域有了基本的了解。同样也感谢大创中一起努力过的小伙伴，是大家的辛苦付出，使我们最终完成了整个项目。接着需要感谢我的各位室友，感谢她们在四年来的支持与鼓励，拥有这样的室友是我的荣幸，希望之后还能一直是朋友。同时，也要感谢默默支持我的两位好友，马和翟。最后，感谢父母家人的宽容与耐心，谢谢你们的鼓励与支持。

**附 录**

空白共八页