廣東工業大學

本科毕业设计（论文）

基于深度学习的企业开放社区用户创意挖掘方法研究

学　　院　管理学院

专　　业　信息管理与信息系统

年级班级　2016 级（1）班

学　　号　3116004232

学生姓名　蔡秀定

指导教师　唐洪婷

2020 年　5 月

基于深度学习的企业开放社区用户创意挖掘方法研究

蔡秀定

管理学院

摘　　要

自 Web 2.0 兴起以来，为了更高效从用户中获得有价值的反馈信息，以提升产品质量，引领创新，国内外知名企业纷纷开始搭建企业开放社区。对于蕴含其中的用户创意，传统的做法通常采用基于机器学习的观点挖掘技术，但这些方法往往存在人工成本高、训练速度慢等缺点。在大数据时代，信息量的爆炸增长更是给这些方法带来了新的严峻性。企业需要一个更高效、即时地从企业开放社区大量的用户帖子中挖掘用户创意的方法。基于这个背景下，本文主要做了以下三个方面的工作：

（1）对于文档粒度的用户帖子主题分类工作，针对传统的单嵌入层卷积神经网络做了一定改进，提出词嵌入层带有 dropout 机制的多嵌入层 CNN 模型，其主要思想是通过结合不同词嵌入层的特点，增强模型的局部语义特征捕捉能力，以此提高分类模型性能。实验表明，词嵌入层加入 dropout 机制有助于模型性能的提升，同时，在 F1 分值指标上，双嵌入层 CNN 模型比传统的单嵌入层 CNN 模型提高了约 1%。

（2）对于句子粒度的用户创意观点文本过滤工作，针对 CNN 模型和 RNN 模型的不足，提出 Transformer 模型和 CNN 模型的组合模型（TF-CNN）。通过与六个基准模型的对比，实验表明 TF-CNN 模型能够充分利用 Transformer 模型能够进行长距离依赖建模的优点和 CNN 模型善于捕捉局部语义信息的特点，从而对包含用户创意观点的句子进行更高效、精确的识别，提高了模型分类性能。

（3）针对大量繁多的可能包含相似创意观点信息的句子集合，在通过均值模型生成句嵌入的基础上，使用基于 Ward Linkage 的层次凝聚聚类方法（HAC），进行试探性的聚类分析工作。在没有使用评价指标的情况下，进行人工审核评估，实验表明 HAC 算法对句嵌入的语义特征具有选取能力，并取得了良好的聚类效果。

本文提出了两种不同粒度下可行的深度学习模型，对企业开放社区中大量的用户自生成文本进行了创意观点挖掘、观点聚类，能够促帮助企业更高效地从开放社区中获取用户创意，灵活反应，保持竞争优势。

关键词：企业开放社区，深度学习，观点挖掘，自注意力机制

ABSTRACT

Since the rise of Web 2.0, in order to get valuable feedback from users more efficiently, improve product quality and lead innovation, well-known enterprises at home and abroad have started to build enterprise open communities. For the user creativity contained therein, traditional methods usually adopt the viewpoint mining technology based on machine learning, but these methods often have the disadvantages of high labor cost and slow training speed. In the times of big data, the explosion of UGC (user generated content) has brought new severity to these methods. Companies need a more efficient and immediate way to tap into the creativity of their users from the large number of user posts in the corporate open community. It is on the following three aspects of work that this thesis, based on the background, researches.

(1) For document granularity users post subject classification work, in view of the traditional single layer embedded convolution neural network made certain improvement, put forward the term embedded layer more embedded with a dropout mechanism model of CNN, its main thought is by combining the characteristic of different embedded word layer, enhance the capacity of local semantic characteristics of the model to capture, in order to improve the performance of classification model. Experiments show that the addition of dropout mechanism to the word embedding layer is helpful to improve the performance of the model. Meanwhile, in terms of F1 score index, the dual-embedding layer CNN model is about 1% better than the traditional single-embedding layer CNN model.

(2) For the text filtering of users' creative ideas in sentence granularity, a combined model of Transformer model and CNN model (TF-CNN) was proposed to address the shortcomings of CNN model and RNN model. Compared with the six benchmark models, the experiment shows that the TF-CNN model can get the utmost out of the advantages of the Transformer model to conduct long-distance dependency modeling and the CNN model is good at capturing local semantic information, so as to carry out more efficient and accurate identification of the sentences containing users' creative ideas and improve the classification performance of the model.

(3) For a large number of sentence sets that may contain similar creative point of view information, on the basis of using the mean value model to generate sentence embedding, a tentative clustering analysis was carried out by using the hierarchical condensation clustering method (HAC) based on Ward Linkage. The experiment shows that the HAC algorithm has the ability to select the semantic features of sentence embedding and achieves a good clustering effect.

In this paper, two feasible deep learning models with different granularity are proposed to mine and cluster the creative ideas of a large number of user-generated texts in the open community of enterprises, which can help enterprises to obtain user ideas from the open community more efficiently, respond flexibly and maintain competitive advantages.

Key words: Enterprise Open Community; Deep Learning; Opinion Mining; Self-attention Mechanism

目　录

第一章　绪论

1.1 研究背景

企业的发展离不开用户反馈。传统的用户反馈获取方式有问卷调查、电话访问、邮件反馈等，但这些方法通常存在效率低、成本高、用户处于被动而消极配合等不足[1]。与传统的获取用户反馈信息方式相比，企业开放社区是基于互联网、面向用户、以用户交流为主企业运营维护为辅的互联网线上交流平台，具有即时高效、用户主动参与、运维成本相对较低等特点。为了挖掘蕴含其中巨大的商业价值、快速找到用户普遍呼吁改进的产品缺陷、有效甄选利于产品更新迭代的用户创意，企业传统的做法是将这些观点挖掘任务交给开放社区的平台运维人员。运维人员进行审核、浏览、答复每一条用户帖子，并从中记录、统计常被用户提及、抱怨或呼吁改进的观点信息，最后反馈给产品相关部门。这种基于人工的观点挖掘方法对于潜藏在用户帖子中有价值的创意观点信息具有非常准确的识别能力，但缺点是效率低、人工成本高，尤其是在今天的大数据时代，企业开放社区中日益的数据增长量给开放社区运维人员带来了巨大的压力。同时，企业不能一味增加运维人员，因为这会增添社区平台的维护成本，这与企业搭建开放平台的初衷之一（降低获取用户创意的成本）是相悖的。因此，有必要使用机器来进行自动观点挖掘。

观点挖掘（Opinion Mining）最早可追溯到 1997 年 McKeown 等人[2]对语义指向的预测的研究。此后的观点挖掘技术通常基于机器学习进行。按挖掘层次进行划分，观点挖掘技术通常可分为文档粒度、句子粒度和方面粒度[3]。文档粒度的观点挖掘通常体现在情感分类任务上，比如说将一篇文档的内容所表现的情感划分为消极、中性和积极[3]。这类任务非常具有挑战性，因为文档中可能存在暗讽、指代等情况，传统的使用情感词典的做法无法很好解决这类问题。同时，一篇文档中可能存在多种用户情绪，因此，独断地对一篇文档的情绪极性进行分类的做法在很多情况下并不合适。句子粒度的观点挖掘通常的做法是进行主观性与非主观性的句子进行分类，在此基础上进行情感分类[5]。方面粒度的观点挖掘则是对句子中的评价对象、对象属性、评价程度、情感极性进行抽取，按任务类型通常可划分为显式方面抽取和隐式方面抽取[6][7] 。基于机器学习的观点

挖掘技术已经取得部分不错的成果，但也存在着特征工程繁杂，只能提取浅层特征等问题。

近年来，深度学习蓬勃发展，各种性能优异的深度学习模型相继被提出，经实践证明，这些深度神经网络模型在一些任务上的表现优于传统的机器学习方法（如决策树、随机森林等）。深度学习技术的出现为传统的基于机器学习的观点挖掘的局限性的解决提供了新的思路，无论是深度学习可提取更深层的特征的优势，还是对大数据集处理的友好性，深度学习技术都让人们看到曙光。综上，如何将深度学习应用于观点挖掘任务具有重要的研究价值和广阔的应用场景。

1.2 研究目标及意义

1.2.1 研究目标

传统基于机器学习的观点挖掘技术对于企业开放社区中日益增长的大量用户自生成内容，存在特征工程复杂、在大数据集上训练速度变慢等问题。对此，本课题的总体研究目标为：基于深度学习技术，对于企业开放社区中大量由用户生成的文本（UGC），完成了不同粒度层次的用户创意观点挖掘算法建模，从而满足企业对于开放社区的用户创意观点自动挖掘的需求。具体研究目标分为：

（1）提出一种文档粒度的用户帖子主题分类算法，从而实现对包含用户创意观点的帖子的识别。

（2）提出一种句子粒度的非创意观点过滤算法，完成对主观、客观句子的区分，从而实现对包含用户创意观点的句子的识别。

（3）提出一种以句嵌入为研究对象的观点自动聚类算法，从而完成相似的观点句子的聚类工作。

1.2.2 研究意义

本课题基于深度学习技术，对于企业开放社区中大量由用户生成的文本（UGC），完成了不同粒度层次的用户创意观点挖掘算法建模，具有以下研究意义。

（1）理论意义

其一，丰富了基于深度学习技术的网络评论文本观点挖掘的研究，尤其是企业开放社区中的用户创意挖掘方面的工作。在文档粒度的观点挖掘任务上，为了适应企业开放社区的文本分布特点，本文没有采取了一般意义上的基于情感极性的方法，而是主张基于话题类型的文档分类方法，并取得了不错的结果，为其他开放社区相关的观点挖掘工作提供了可参考的研究思路。

其二，提出了一种基于组合思路的文本特征提取器。对于现自然语言处理领域流行的几种常见的特征提取器：RNN、CNN、Transformer 等，本文将 Transformer 的编码器模块单独分离出来作为 CNN 的前置特征提取器，提出 TF-CNN 模型，实验表明 TF-CNN 具有更强的特征表征能力。从某种程度来说，TF-CNN 可充当其他自然语言处理下游任务的特征提取器，应用场景广泛，具有通用意义

（2）现实意义

其一，本文设计了一种高效的用户创意观点挖掘算法，可大大减轻开放社区平台运维人员的工作负担。前期只需要部分的数据标注工作，后期只需要投入少部分人力，即可由机器自动完成 UGC 中的用户创意观点挖掘工作。文本过滤、文本提取等工作由机器自动完成的意义在于从一定程度上减少了运维人员的重复性低价值密度工作，从而让运维人员可以将更多的精力投入到更高层次的工作，如用户创意合理性判断、创意观点聚合提取等。这些工作更人性化，更利于员工创造出更高的价值。

其二，高效即时从企业开放社区中挖掘用户创意观点有利于企业保持对市场的敏感性。通过高效获取开放社区中的用户反馈信息，听取用户的意见和建议，并迅速对产品做出调整，敏捷开发，从而为产品赋予了经久不衰的活力。

1.3 论文主要工作

基于研究目标，本课题的主体工作内容为首先准备数据集，并训练好相应的词向量模型，在此基础上构建文档粒度的主题分类模型、句子粒度的非创意观点过滤模型，通过超参数优化实验，以此找到适合模型的较佳参数，从而提高模型性能，最后设定评价指标，通过与基准模型的对比实验以验证模型的有效性。在这两项工作完成后，最终对非创意观点过滤得到的结果进行自动聚类分析，得用户创意观点簇。整体研究框架如图 1-1 所示。

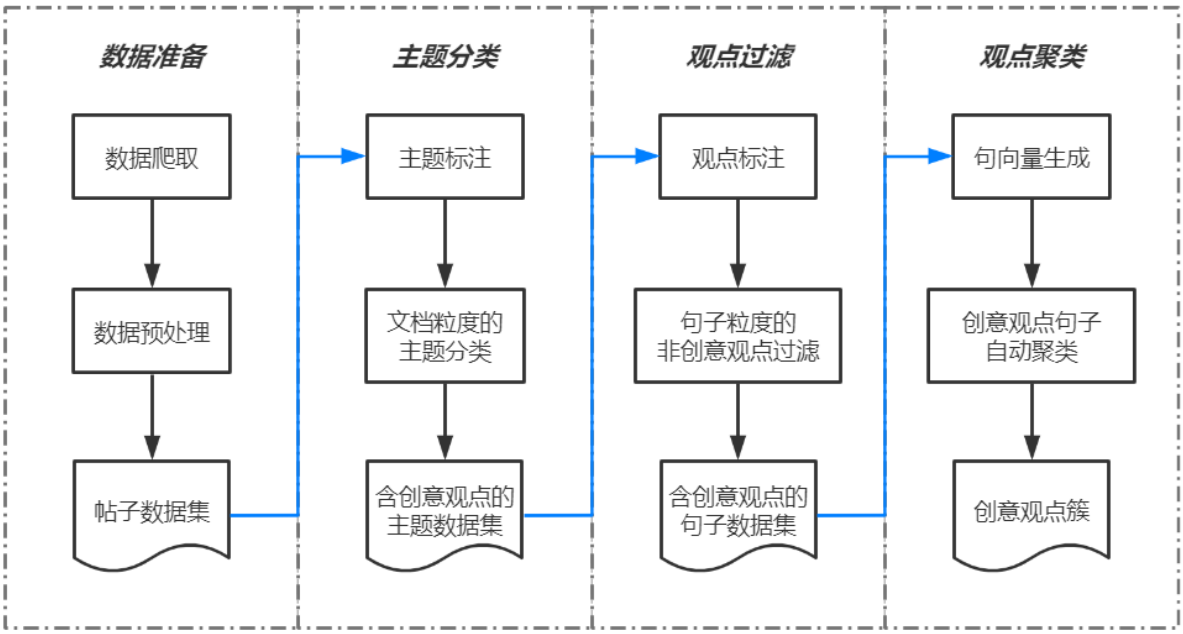


图 1-1 研究路线图

本课题具体做了以下四个方面的工作：

（1）在分析并选择了国内合适的企业开放社区作为本课题的研究对象的基础上，通过编写爬虫程序，爬取了花粉社区 2020 年 1 月到 2 月 Mate30 手机的用户帖子，并进行数据清洗工作。

（2）介绍了 CNN 的模型结构及其原理，并在此基础上提出加入词嵌入层 dropout 机制的多嵌入层 CNN，应用于文档粒度的主题分类任务。在具体实验中，通过超参数实验的结果，得到较佳的参数设置。在此基础上，通过与单嵌入层 CNN 的对比，实验并验证了带有词嵌入层 dropout 机制的多嵌入层 CNN 能取得更好的分类模型性能。

（3）对于句子粒度的文本分类任务，总结了常见的深度神经网络模型及其不足，并在此基础上提出 Transformer 和 CNN 的组合模型（TF-CNN）。并通过实验完成非用户创意观点过滤任务，通过与多个基准模型的对比实验，验证了 TF-CNN 模型能够发掘更丰富、更深层次的语义信息。

（4）对于用户创意观点聚类任务，提出基于 Ward Linkage 的层次凝聚聚类算法，使用均值模型生成句嵌入，进行了试探性的聚类分析工作，取得不错的聚类结果。

1.4 创新性工作说明

本课题探讨了基于深度学习的企业开放社区用户创意观点挖掘方法，在已有的前人研究基础上，主要开展了以下创新性工作：

（1）针对文档粒度的主题分类任务，设计并实现了一种新模型，即词嵌入层带 dropout 机制的多嵌入层 CNN 模型。其创新点在于，其一，传统的单嵌入层 CNN 存在语义特征表达单一的不足，多嵌入层的设计可以有效改善这一点；其二，在词嵌入层加入 dropout 机制，从一定程度上可减少过拟合情况的发生，从而增强模型的泛化能力，提高分类模型性能。

（2）针对句子粒度的非创意观点过滤任务，设计并实现了一种新模型，即 TF-CNN 模型。其创新点在于，对于现 NLP 领域流行的三大特征提取器：RNN、CNN、Transformer，提出了基于组合思路的“Transformer+CNN”特征提取器，从而实现在完成长距离依赖关系建模的基础上进行局部语义特征捕捉，大大增强了对文本特征的捕捉能力。

（3）完成了基于 Ward Linkage 的层次凝聚聚类算法的新应用，即企业开放社区中的用户创意观点自动聚类。其创新点在于，一方面可以进一步减轻社区运维人员的工作负担；另一方面，验证了聚类算法应用于句嵌入的可行性，为后人的进一步工作提供了思路。

综上所述，本课题提出的模型与方法为企业开放社区中的大量用户自生成内容提供了一种高效的创意观点挖掘思路与实现，可以有效减轻社区运维人员的工作压力，协助企业降低用户创意的获取成本、提高对市场的敏感性，从而进一步扩大竞争优势；同时也为网络评论观点挖掘相关问题的研究提供了一定的借鉴价值和参考思路。

第二章　国内外研究和发展现状

自 Web 2.0 兴起以来, 为了更高效即时从用户中获得有价值的反馈信息，以提升产品质量，引领创新，国内外知名企业纷纷开始搭建企业开放社区。国外企业如戴尔开设线上平台[1]以征求全球用户的意见或建议，乐高玩具通过创享空间[2]让玩家分享他们设计和创造的作品图片以激励用户，并从中创新。国内企业的例子也陈举不乏，如海尔可开设智家社群[3]，吸引全球企业、用户进行交互式创新，不断从中产出爆款。智能手机品牌如华为、小米、Vivo 等都有自己开放社区，且其产品大部分的更新迭代动机就来自其社区。企业开放社区的迅速发展，引起了国内外学者的关注和研究。但相对传统互联网平台如微博、淘宝等而言，企业开放社区的观点挖掘研究起步较晚，目前仍存在许多困难和问题，具有很大的探索空间。本章将介绍企业开放社区观点挖掘方面已有的研究，并跟踪深度学习在自然语言处理方面的部分研究进展，随后对聚类分析进行概述，最后引出二分类评价指标。

2.1 企业开放社区观点挖掘相关研究

企业开放社区中用户创意内容主要来自于用户在社区上发布的帖子。这种帖子具有短文本的特点，一般在 10 到 200 个字左右。与微博等传统短文本相比，开放社区的短文本具有一些独特之处：首先，开放社区的短文本是用户为向社区管理者倾诉或与其他社区参与者分享而被用户生成的，文本内容更丰富，目的性更强。其次，微博内容涉及范围非常广，涵盖生活、社会、经济、政治等，而开放社区的短文本的话题则相对专一，主要围绕企业所提供的产品、服务而展开。当然，他们也具有共同点，其一，文本内容长度相对较短，一般在 200 字以下；其二，内容长度虽短，但语义信息繁杂。并且可以通过 emoji 表情、话题标签、图片、短视频等形式表达传递更丰富的信息和情感；其三，都具有实时性，比如说微博用户讨论的话题与时事热点相关（如口罩的正确戴法），开放社区用户讨论的话题与当前企业产品、服务的最新迭代版本相关（如手机系统的安全补丁）；其四，用户活跃，文本更新迭代速度快。在这里选择微博短文本与开放社区的

[1] www.delltechnologies.com

[2] https://www.lego.com/en-us/createandshare

[3] https://bbs.haier.com/

短文本对比，其原因是目前在企业开放社区短文本上的研究尚在起步阶段，而微博短文本的研究方法相对成熟[8]。虽然二者不完全相同，但存在一定相似共通性。对微博短文本的分析一般涉及文本表示方法、情感分析方法、文本分类方法等。考虑到本课题的研究内容，以下将介绍文本表示方法和文本分类方法。

2.1.1 文本表示方法

计算机不能直接理解字符串形式的文本信息，同时为了便于后面的处理和计算，作为输入的原始文本需要转换为等尺度的数值型向量或矩阵表示。

布尔模型[9]是一种较常见的文本表示方法，亦称独热编码（One-Hot Representation）。该模型首先根据语料集合去重得到一个词表，设词表规模为 N，生成一个 N 阶的单位矩阵，记为 $(\alpha_1; \alpha_2; \dots; \alpha_N)$，其中 α_i $(i = 1,2, \dots N)$ 是 1×N 的行向量，则词表中的第 i 个词可用 α_i 表示。布尔模型也可用于生成句表示，即将该句子经分词后的每个词对应词汇表上的维度设置为 1 即可。当显然这存在一个明显的问题，即不考虑词序信息，如“水动风凉夏日长”和“长日夏凉风动水”的独热编码是一样。同时，布尔模型无法进行词间相似性度量；此外，布尔模型还存在一个致命缺陷：维数灾难，一旦训练语料足够大，则对应建立的词汇表的词向量维度是一个天文数字。

为了解决传统文本表示方法的多个缺陷，Bengio 等人[10]在 2003 年将神经网络加入语言模型（Language Model）并进行训练，其间可以生成低维度的词向量模型。此后有更多的学者开始关注分布式词向量模型，并从更丰富的语义、更高效的训练过程等方面改进词向量模型[11][12]。词向量模型在自然语言处理领域真正成为文本表示方法的主流是在 2013 年，Mikolov[13][14]基于大规模训练语料，使用循环神经网络语言模型高效快速训练出词向量模型，该模型比以往的词向量模型的语义表征能力都强，并以开源的方式发布 Word2vec 工具。Word2Vec 工具包含两种词向量训练模型，一是 CBOW 模型，它通过预测词的上下文来进行预测，如“明天__下雨”，“__”处表示预测词，此处可能预测“可能”；另一种是 SG 模型，通过给定词来预测给定词得到上下文词。相对于 CBOW 模型而言 SG 模型需要更多的训练语料，因为它是给定很少的信息而进行预测更多的信息，这需要更大的训练语料支撑。在这两个模型实际训练过程中，为了降低训练时间复杂度，往往会配合不同的训练策略。常见的训练策略[14]有 Hierarchical Softmax 和 Negative

Sample。另一个常使用的词嵌入学习手段是 GloVe（Global Vector），它可以针对全局词-词共现矩阵的非零项进行训练[15]，从而往词向量模型中加入全局语义信息。

2.1.2 文本分类方法

分类问题无论在哪个领域都是研究人员关注的热点，文本分类也不例外。文本分类技术最早可追溯到 20 世纪 50 年代，经过几十年的研究发展，文本分类已经经历了从基于规则，到机器学习，再到深度学习等三个阶段。

早期的文本分类主要是基于规则来实现，而且对人员有极高的要求：通常需要是领域内的专家以及对问题有深刻的理解。对于复杂的模型，通常需要设计繁多的规则来实现，基于专家规则的文本分类效率低且容易出错。上个世纪 80 年代，人们开始通过知识工程来建立专家系统，专家系统通常包括知识系统和推理系统两大部分，尽管取得一定性能的提升，但本质上仍属于基于规则实现的层次。到了 90 年代，网上冲浪的流行产生了海量的在线文本，同时机器学习技术兴起，研究者开始再次重点关注文本分类领域，感知机[16]，K 近邻[17]，朴素贝叶斯[18]，支持向量机（SVM）[19]等经典机器学习模型被应用于文本分类。这类技术取得了不错的成果，但这些算法在真正建模前一般还需要进行特征工程，这项繁杂的工作通常由人工进行，手动设计或组合。也正是因为如此，经过特征工程得到的特征通常具有主观局限性。此外，特征工程无法提出较深层的特征，从而致使分类器过拟合。这些问题随着近年来深度学习的出现而得到解决，引领来文本分类划时代的另一个热潮。

2014 年， Kim[20]提出 TextCNN 模型，首次将卷积神经网络应用到文本分类。

Pengfei Liu[21]等人在 2016 年提出了 TextRNN 模型。该模型的提出是为了改进 TextCNN 的固定卷积窗口而无法建模更长的序列信息的问题。

Yang Z[22]等人尝试将 Attention 机制引入文本分类问题而提出了 TextCNN+Attention 模型。引入层次 Attention 机制后，显著提高了模型的可解释性，能够直观的解释各个句子和词对分类类别的重要性。

Johnson 等人[37]对于 CNN 无法提取全局语义表征的局限性，提出了深度金字塔 CNN 模型。该模型具有计算开销增量小，可捕捉全局语义信息等优点。

2.2 深度学习研究现状

深度学习是机器学习领域的浅层神经网络的延伸，其特点在于“深”。传统的神经网络技术观点认为，数量少而精（一层或两层）和少量数据的神经网络模型更加实用有效。而深度学习的神经网络模型打破了这一传统观点，它可以整合、集中、利用更多的神经网络的学习（表示）能力，以取得更好的性能效果。自 2012 年以来，深度学习模型在各个领域，尤其是机器视觉和自然语言处理两大领域皆取得了不俗的效果，表现出出巨大的上升潜力。

2.2.1 卷积神经网络

卷积神经网络（Convolutional Neural Networks，CNN）是一类特殊的前馈神经网络，由多层神经网络构成，其最大的两个特性是具有局部感受野和权值共享性：前者意味着 CNN 拥有局部特征的抽取能力，这种特征可以是人类可直接观察的浅层特征，也可以是人类无法理解的更加抽象的深层特征；后者则是 CNN 区分于传统的全连接网络的一大优势，模型通过参数共享，大大降低了网络参数的数量级，提高了模型的时间性能。

卷积神经网络的相关研究最早可追溯到 1968 年[23]，但因为高开销的计算时间成本，CNN 的发展一直缓慢[24][25]。直到 2011 年，吴恩达团队发现图形处理单元（GPU）可用于加速深度神经网络的训练，原本需要几周的训练时间缩短至几天，此后研究人员争先仿效[26]。2012 年，Krizhevsky[27]等人提出 AlexNet 模型，在 LSVRC 比赛中将以往最优成绩的错误率降低了近 50%，取得冠军，体现了 CNN 性能的重大突破。此后，CNN 开始蓬勃发展。从本质上讲，CNN 结构的天然可并行计算性和 GPU 等硬件的发展和支持是 CNN 能够再次兴起的关键因素。另一方面，因为 CNN 的局部检测与 N-gram 相似，因此 CNN 在 NLP 领域也受到了广泛关注。2014 年，Kim[20]首次将 CNN 模型用于解决文本分类问题，在部分数据集中，取得了优于传统机器学习算法（如 SVM）的效果，拉开了 CNN 模型在自然语言应用领域的序幕。

2.2.2 循环神经网络

循环神经网络（Recurrent Neural Network，RNN）最早由 Mikolov 等人[28]提出，可以有效解决 CNN 没有“记忆能力”的问题，对于处理存在依赖关系的序列数据特别有

效。RNN 的主要思想是将前一个输入的隐藏层的输出值或输出层的输出层保存，并作为下一个输入的参数，也正是这样的设计使其具有短期的“记忆性”。之所以说 RNN 的“记忆性”是短期的，是因为网络层数过深的 RNN 模型在训练过程中会出现梯度爆炸或消失的情况，导致模型无法再进行学习。这个缺陷使得 RNN 模型可以处理任意长度的序列数据显得不切实际。针对标准 RNN 模型无法进行长距离回溯以及一个元素可能依赖于后一个元素的问题，Hochreiter 等人[29]提出了 Bi-RNN 模型（Bidirectional RNN）。Bi-RNN 模型的本质是对于一个序列信息用同一个 RNN 模型训练两次，不同之处在于一次是顺序，一次是逆序，并将结果一起传给输出层。因此，Bi-RNN 模型考虑了更丰富的语义信息：一个元素的前一个元素和后一个元素。

2.2.3 长短时记忆模型

长短期记忆网络（Long Short Term Memory）[29]是另外一种为解决 RNN 不适用于回溯距离短的问题的特殊类型的 RNN，能够完成长距离语义依赖关系建模。相比于传统的 RNN 模型，LSTM 的结构更加复杂，由四种门控制其状态，从而能取得较之 RNN 更佳的模型性能。

传统 LSTM 只能在顺序数据上起作用，Tai 等人[30]将 LSTM 推广到树结构 LSTM（Tree-structured LSTM，Tree-LSTM），并在表示句子意义方面表现出比顺序 LSTM 更好的性能。LSTM 的另一个变体是门控递归单元（GRU）[31][32]，它在标准 LSTM 的结构上进行了改进，简化了其结构，并越来越受到欢迎。

2.2.4 Transformer

Transformer 一种特殊 Seq2Seq 模型，常用于代替 RNN 模型，由 Vaswani 等人[33]在 2017 年提出。一般而言，Seq2Seq 模型需要内嵌 CNN 或 RNN 以运作，Transformer 的不同之处在于使用了自注意力机制（Self-attention Mechanism，SAM）代替了 RNN 结构，并在开始提出的时候在机器翻译任务上取得 STOA 效果，随后在 2018 年因取得多个 NLP 任务的 STOA 而声名大噪 Bert 模型[39]也是基于 Transformer 实现。如图 2-1 所示，Transformer 总体由编码器模块（见图左）和解码器模块部分（见图右）两个模块构成。

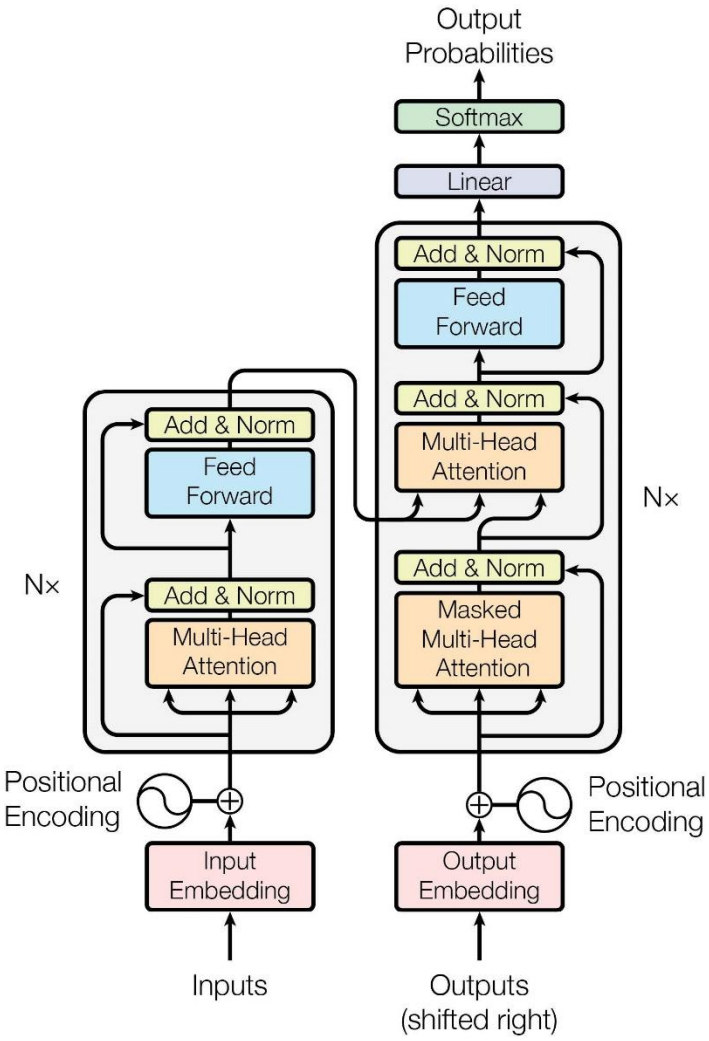


图 2-1 Transformer 架构[33]

编码器模块和解码器模块分别由 N 个编码器和 N 个解码器组成。当数据进入 Input Embedding 层并与 Positional Encoding 层结合，开始进入第一个编码器，第二个编码器接受第一个编码器的输出作为输入，直到数据流经 N 个编码器，最后一个编码器将结果传递给解码器模块的每个解码器中，具体过程如图 2-2 所示。

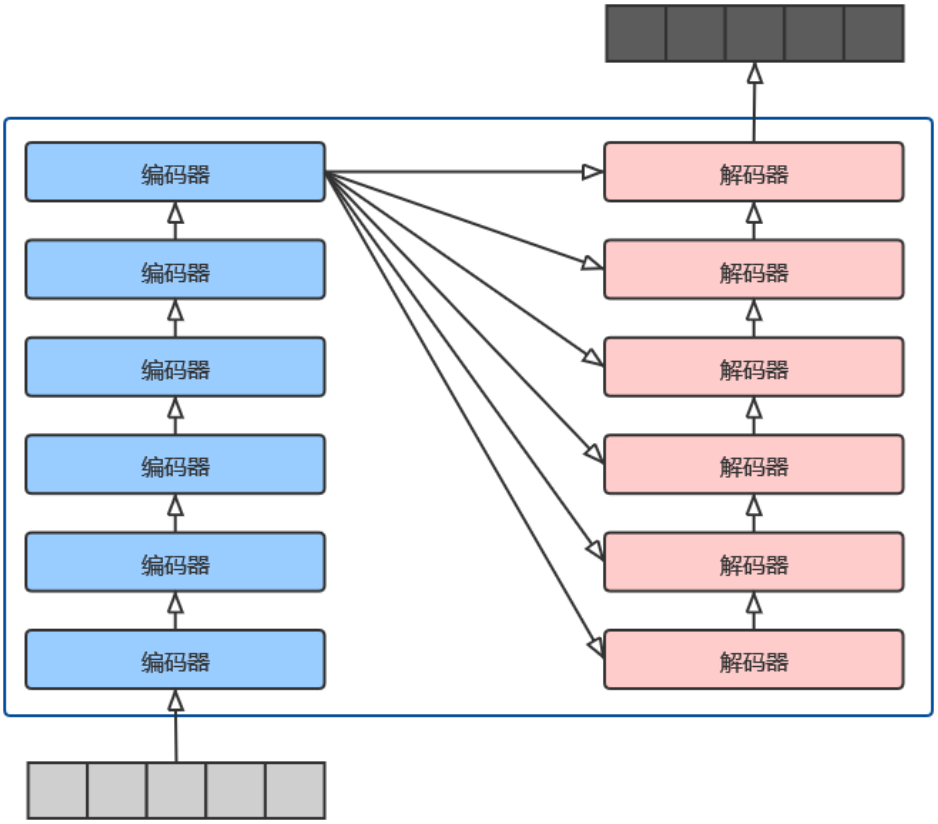


图 2-2 Transformer 的编码器和解码器

2.3 聚类分析研究现状

聚类分析不同于分类，它是基于某种方法，经过无监督学习将无标签数据集自动划分成多个簇的过程。一般要求簇间对象的相异度较大，而簇内对象的特征较相似。聚类作为统计学的一个传统研究领域，经过数十年的发展，诞生了许多分别适用不同任务的聚类算法。按研究对象或应用场景，常见的聚类分析大致有以下几种：

（1）划分方法：对于任意一个元素数量为 N 的元素集合，我们每次将这个集合分割成 K 个区域（K 小于等于 N），也就是 K 个簇。在经过第一次迭代的随机 K 个区域的初始化后，按照一定的判别方法，重新分配每个对象所属的区域，直到所有对象所属的区域不再发生变化。该方法适用于中小型数据集，对于大型数据集，有着时间复杂度高的缺陷，此外，在训练前需要了解数据的分布，以此设定簇数 K。此间具有代表性的聚类算法有 K 均值算法、K 中位数算法等。

（2）基于密度的方法：这种聚类方法的研究思路是使用元素的密度作为簇划分标准，当一个区域的元素密度超过一定上限，则视为一个簇。基于密度的聚类算法对元素在空间中的分布的疏密性敏感的，因此它可以发现不规则形状聚集的簇；但当元素在空间中分布稀疏或均匀时，很难发挥效果。这类算法著名的有 DBSCAN 算法。

（3）层次方法：根据层次进行的方向，可分为分解与凝聚两类。以层次凝聚聚类方法为例，首先将拥有 N 个元素的元素集合分别初始化为 N 个簇，然后依据特定的簇间度量距离方法度量两两个簇之间的距离，常见度量方法有最短距离发、最长距离法、Ward 距离法等，然后合并簇间距离最短的两个簇。依次迭代，直到所有簇都合并为一个簇，或提前减少到所需满足的簇数。层次分解聚类方法则相反，首先将整个元素集合初始化为一个簇，然后按照一定的距离度量方法，从中分解出一个新簇。直到整个簇分解为 N 个簇，或提前增加到所需满足的簇数。层次聚类算法的计算开销相对较小，也无需提前指定聚类簇数（如果不需要），但存在一个很明显的问题：无法调整错误，聚类性能的好坏与距离度量方法有关。具有代表性的有：聚类特征数算法（CF-Tree）、基于层次结构的平衡迭代聚类方法（BIRCH）等。

（4）基于网格的方法：将元素分布的空间切分为若干个空间格子，使其呈网格状，此后的聚类过程在网格结构中进行。这种聚类算法具有无视元素数量、处理效率高等特点，因为处理时间仅和划分的网格数有关，而与元素集合的规模无关。常见的有：STING 算法、MMNG 算法、ENCLUS 算法等。

2.4 评价标准

本课题实验主要涉及文本分类任务，为评估模型性能，采用以下评价标准。

（1） 准确率

准确率是所有预测正样本无误（记为 TP）和预测负样本无误（记为 TN）占总预测数量（记为 S）的比重，公式为：

$$Accuracy = \frac{TP + TN}{S} \tag{2.1}$$

（2） 精确率

精确率用于表示“虚警”的程度，是预测正样本无误（TP）占全部预测为正类别的占比。记将负样本预测为正样本为 FP，则其定义为：

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

（3） 召回率

召回率用于表示“漏报”的程度，是预测正样本无误的在实际总正样本中的占比。记将正样本预测为负样本为 FN，则有：

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

（4） F1 值

F1 值同时考虑精确率和召回率，即在“虚警”和“漏报”间做一个权衡。F1 值的取值范围为 $[0, 1]$，值越大，模型分类性能越好。公式为：

$$F_1 = \frac{Precision \cdot Recall}{0.5 \cdot (Precision + Recal)} \tag{2.4}$$

2.5 本章小结

本章主要介绍了四部分：企业开放社区观点挖掘相关研究、深度学习研究现状、聚类分析研究现状以及二分类评价标准。第一部分主要介绍了两种文本表示方法（包括独热编码和词嵌入）和文本分类研究现状（经历了从基于规则到机器学习，再到深度学习技术的发展）。第二部分则依次介绍在自然语言处理领域较为活跃的深度神经网络模型，包括：CNN、RNN、Bi-RNN、LSTM 等，以及最后介绍了 Transformer 模型。第三部分概要介绍了常见的聚类算法类型以及各自的特点。本章最后介绍了二分类模型常见的性能评价指标。本章综述了与课题相关的理论知识和研究进展，为后续工作的进行奠定了坚实的理论基础。

第三章　数据集与词向量模型相关介绍

本章首先概要介绍了数据集的选择，然后从各个数据集获取的角度出发，介绍了数据预处理的主要思路与过程，在本章最后阐述了词向量模型训练的过程。总的来说，本章的工作为后续章节的模型实验奠定坚实的基础。

3.1 数据集选择

本课题的目标是从企业开放社区的由用户生成的内容中挖掘创新观点。总体任务流程如图 3-1 所示。

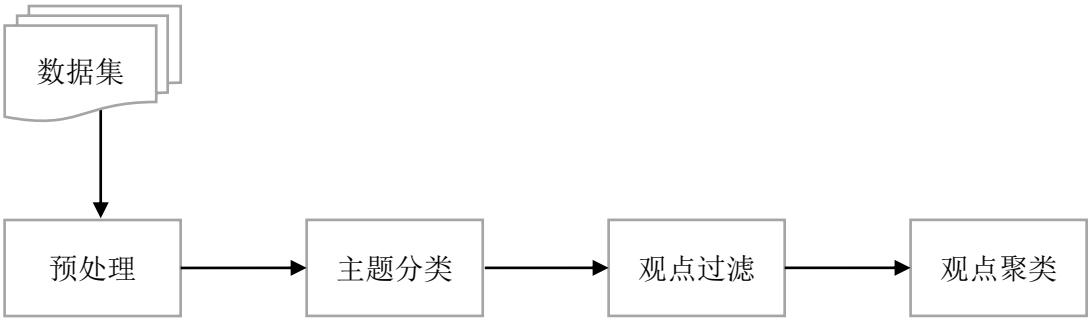


图 3-1 总体任务流程

从整个任务流程来看，本课题用到两种类型的数据集，一是用于模型训练、验证、测试的研究对象的数据集；二是用于词向量模型训练的数据集。

对于研究对象（即企业开放社区）的数据集，本课题选取华为手机的花粉社区作为研究对象，考虑有二，一是其用户基群大[34]，二是华为花粉社区的活跃度颇高，以华为在 2019 年 9 月发布的 Mate 30 系列为例，目前这个板块的用户日均发帖数在一万左右。这个数据意味着，一方面，该产品受到用户的极大欢迎，用户积极参与产品评论；另一方面，也正是产品仍存在许多不足，用户积极参与评论反馈，以呼吁企业改进产品。从上述几点看来，华为手机的花粉社区是本课题较为理想的研究对象。

对于用作词向量模型训练的数据集，该类数据集的选取有两个要求，第一，数据集要足够大；第二，该数据集的用语习惯与研究对象数据集的不能相去甚远。花粉社区数

据集自然是作为词向量向量的首选，但其本身数据量并不大，同时鉴于互联网上尚无开源可用的大型企业社区数据集，本课题最终选取中文维基百科[4]作为补充语料。

3.2 数据预处理

这里给出两个数据集的数据预处理总流程图，见图 3-2。

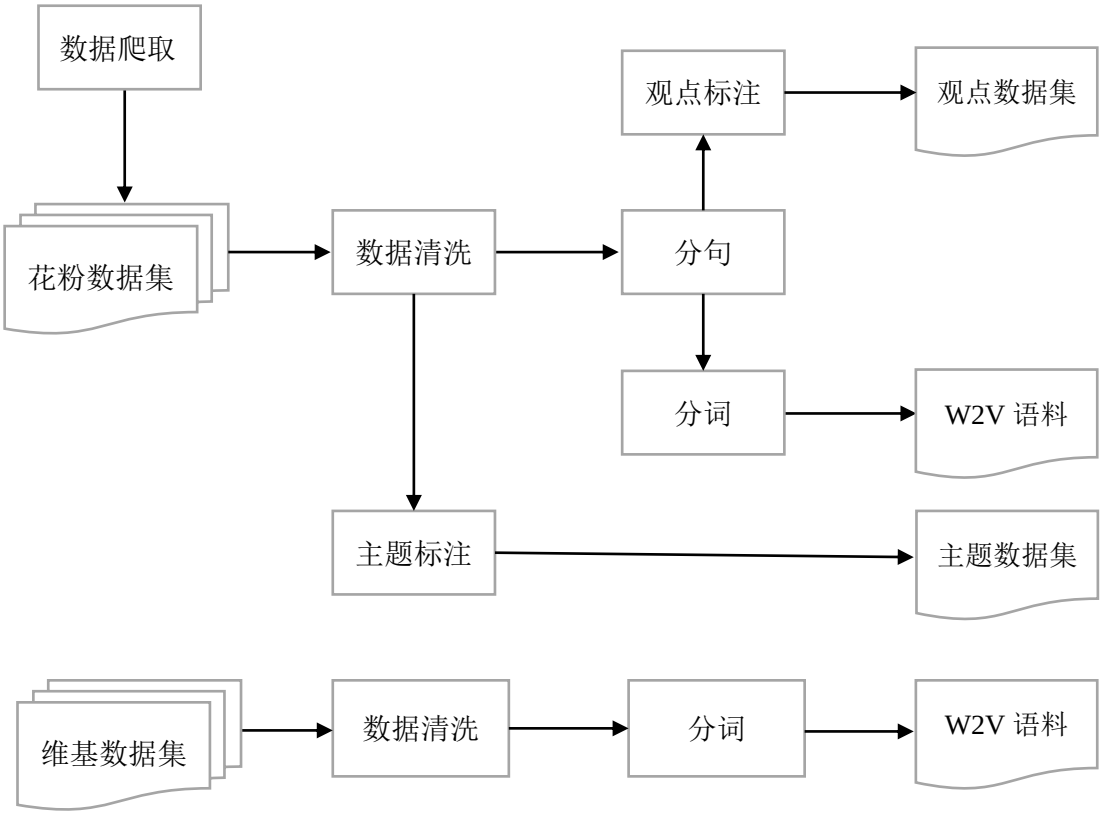


图 3-2 数据集处理总流程

下面先简单阐述数据清洗部分，然后按最终得到的数据集给出花粉数据集处理的主要步骤。关于中文维基数据集，其大部分处理流程（数据清洗、分词等）和花粉数据集的相似，故不作展开。

[4] https://dumps.wikimedia.org/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2

3.2.1 数据清洗

本课题主要爬取了花粉社区的 Mate30 手机板块 2020 年 1 月到 2020 年 2 月的部分数据，丢弃空白、重复项，共得到 48917 条数据（即 48917 个用户帖子），组成花粉数据集。该数据集由用户所发的帖子的标题、内容及其他方面要素（如帖子浏览量、发帖时间等）构成。

由于数据集来自于类似于论坛的企业开放社区平台，其中包含了许多不规范、对课题研究没有参考价值的字符串，如颜文字、URL 链接、HTML 标签等。对此，在将数据投入模型训练前，需要对数据集进行一系列预处理，具体包括：

（1）形式归一化。一个词语可能有多个形式，但表达的内涵是一样的，比如“華為”和“华为”，“Mate Pro”和“mate pro”，“P30”和“Ｐ３０”。对此，需要对原始文本依次进行简繁体转换、大小写统一，以及全半角转换。

（2）主题无关字符串过滤。爬虫得到的原始数据中难免包含形如 HTML 标签、社区运维信息（如：游客请登录）等与用户帖子不存在上下文语义联系的字符串，该类型字符串需要清除。以及对于图像类型数据，爬虫在爬取时，仅采集其链接地址，因为本课题仅涉及到文本类型的数据，该类型 URL 字符串对课题研究的主题不具备参考价值，所以也需要进行正则过滤。

对花粉数据集进行数据清洗后，我们过滤掉了大部分的文本噪声。这一步的工作为后续的分句、分词、词向量训练等工作提供了比较顺利的基础。

3.2.2 主题数据集

在数据爬取部分我们已经爬取了每个用户帖子的主题类型，但用户帖子原有的主题类型可能出现标签错误的情况，这里按主题类型层次抽样了 4000 条数据由两个人独立进行审查，并对主题标签错误的帖子给予纠正。这里给出一个示例，见表 3-1、表 3-2。

表 3-1 主题标注前

帖子内容	主题类型	状态
系统。现在最新的系统怎么样，有没有大变化？	玩机技巧	标签错误
电子书模式始终没有，官方也没有消息，心很凉	功能建议	标签正确

表 3-2 主题标注后

帖子内容	主题类型	状态
系统。现在最新的系统怎么样，有没有大变化？	分享交流	已纠正
电子书模式始终没有，官方也没有消息，心很凉	功能建议	不变

3.2.3 观点数据集

（1）分句

在观点标注前需要进行中文分句工作。这里的分句工作是指将一个用户帖子（通常是由多个句子组成的段落）切分成若干个句子。分句的工作相对简单，本课题使用基于规则的正则表达式进行，即识别目标文本中的换行符和断句符号（如“。！？!?”等）。

（2）观点标注

在对花粉数据集中 48917 个用户帖子进行句子切分后，共得到 184632 个句子。随后从中随机抽取 5000 个句子由两个人独立进行标注，将标注结果合并，作为后续非用户创意观点信息过滤模型的训练数据。

这里给出一个分句及观点标注示例，见表 3-3。

表 3-3 分句、观点标注示例

分句前	一句“小艺小艺”家里顿时热闹了。以前 mate9pro 都可以设置自定义唤醒词，现在华为产品为何全是强制欢迎词“小艺小艺”，非常不方便，希望能改回能自定义唤醒词！支持的朋友顶起来！
分句后	一句“小艺小艺”家里顿时热闹了。// 以前 mate9pro 都可以设置自定义唤醒词，现在华为产品为何全是强制欢迎词“小艺小艺”，非常不方便，希望能改回能自定义唤醒词！// 支持的朋友顶起来！//
观点标注	一句“小艺小艺”家里顿时热闹了。[非创意观点信息] 以前 mate9pro 都可以设置自定义唤醒词，现在华为产品为何全是强制欢迎词“小艺小艺”，非常不方便，希望能改回能自定义唤醒词！[创意观点信息] 支持的朋友顶起来！[非创意观点信息]

3.2.4 词向量语料

进行词向量模型训练的前置条件是对参与词向量训练的语料进行分词，且分词结果的质量会影响模型的最终结果。

中文分词技术目前在国内较为成熟，且已有很多开源的中文分词库，如：Jieba，SnowNLP，HanLP 等。本课题采用目前较为流行的基于 Python 语言的 Jieba 分词库作为分词工具，利用该库的精确分词模式来完成分词步骤。因为在实际操作中，这两个数据集的分词处理有一定区别，这里做一点说明。花粉数据集属于手机社区的语料，存在不少手机圈子的新兴词汇（圈子用语，如曲面屏），这些词汇，如果单纯依赖于 Jieba 自带的原生词典，是很难识别出来的。所以有必要建立花粉数据集的用户词典，且该用户词典的质量会直接影响最终分词的效果。为尽可能完善花粉数据集的用户词典，本课题采用了整合互联网上手机相关词典并配合正则匹配的新词发现方法。而对于中文维基数据集，分词使用的仍是 Jieba 自带的词典。

在对两个数据集完成分词操作后，可得到“花粉语料”和“维基语料”。

下面给出一个分词示例（词间用“/”隔开），见表 3-4。

表 3-4 分词示例

分词前	分词后
一句“小艺小艺”家里顿时热闹了。	一句/小艺/小艺/家里/顿时/热闹/了
以前 mate9pro 都可以设置自定义唤醒词，现在华为产品为何全是强制欢迎词“小艺小艺”，非常不方便，希望能改回能自定义唤醒词！	以前/mate9pro/都/可以/设置/自定义/唤醒词/现在/华为/产品/为何/全是/强制/欢迎词/小艺/小艺/非常/不/方便/希望/能/改回/能/自定义/唤醒词
支持的朋友顶起来！	支持/的/朋友/顶/起来

3.3 词嵌入表示

在这一节将训练多种词向量模型（分别基于“花粉语料”、“花粉语料+维基语料”两个语料等），作为后续实验的词嵌入表示。

3.3.1 实验相关说明

（1）实验环境

如表 3-5 所示。

表 3-5 实验环境

实验环境	具体配置
操作系统	Windows 10 Education 1909
CPU	Intel(R) Core(TM) i7-7700HQ CPU @2.80GHz
内存	16 GB
编程语言	Python 3.7
词向量训练工具	Gensim 3.8.1
分词工具	Jieba 0.42.1

（2）词向量训练工具

本课题主要使用 Python 的一个开源包 Gensim 进行词向量训练。Gensim 训练词向量时主要的参数设置及每个参数的意义如下：

（1）sentences：即上一节经过 Jieba 分词得到的句子列表，每一项数据都是由分词后的词语构成的句子。

（2）sg：模式参数，可选 0 或 1，分别表示的是 Word2vec 的两个模型，CBOW 模型和 Skip-Gram 模型。

（3）size：目标词向量模型维度。

（4）window：即预测滑动窗口大小，本文实验设置的参数是 5，即每个词考虑上下文前 5 个词和后 5 个词。

（5）min_count：最低词频数，当语料中词语出现的次数低于该参数，则会被过滤。本文设置的参数为 5。

（6）workers：模型训练时并行化数量。这里根据实验环境的 CPU 核数确定其值。

3.3.3 实验结果

花粉语料和维基语料在体积上差别很大，前者大概有 10MB，后者的体积则将近 900MB。在第二章中的 2.1.1 节中介绍了 CBOW 模型对于较小的数据集是有效的，而 SG 模型则更适合于较大的数据集。因此，将采用 Word2vec 的两种训练模型分别对两个语料进行词向量训练，如表 3-6 所示，可得到三种预训练词向量模型，其中 RAND 是作为后续对比实验的随机生成词向量模型。此外，在词向量模型的维度上给予控制，分别设置 100 维和 200 维，因此最终可得到 9 个词向量模型。

表 3-6 词向量训练结果

词向量模型	训练语料	训练方法	维度
CBOW_300D	花粉	CBOW	300
SG_300D	花粉+维基	SG	300
RAND_300D	无	随机生成	300

3.4 本章小结

本章主要介绍了数据集的获取以及语料数据转化为词向量模型的过程。首先开门见山介绍了课题的总体任务流程，并扼要说明了本课题的研究对象为花粉社区。接着介绍了三个数据集的预处理流程，具体过程包括数据清洗、分句、标注、分词等。最后通过词向量训练工具 Gensim 训练得到六个预训练词向量模型和随机生成了三个词向量模型，为后续章节的实验提供了坚实的基础。

第四章　基于卷积神经网络的用户帖子主题分类

4.1 问题描述

本章的任务是进行文档粒度（即用户帖子）的用户创意观点挖掘。尽管大部分企业开放社区强制要求用户在发布新帖子的时候选择帖子的话题类型，以华为花粉社区的Mate30 手机板块为例，如图 4-1 所示，用户在发布新帖子时需要主动选择话题（主题）类型：分享交流、iOS 换机专栏、玩机技巧、功能建议、问题反馈、晒单评测、应用资源、其他等。

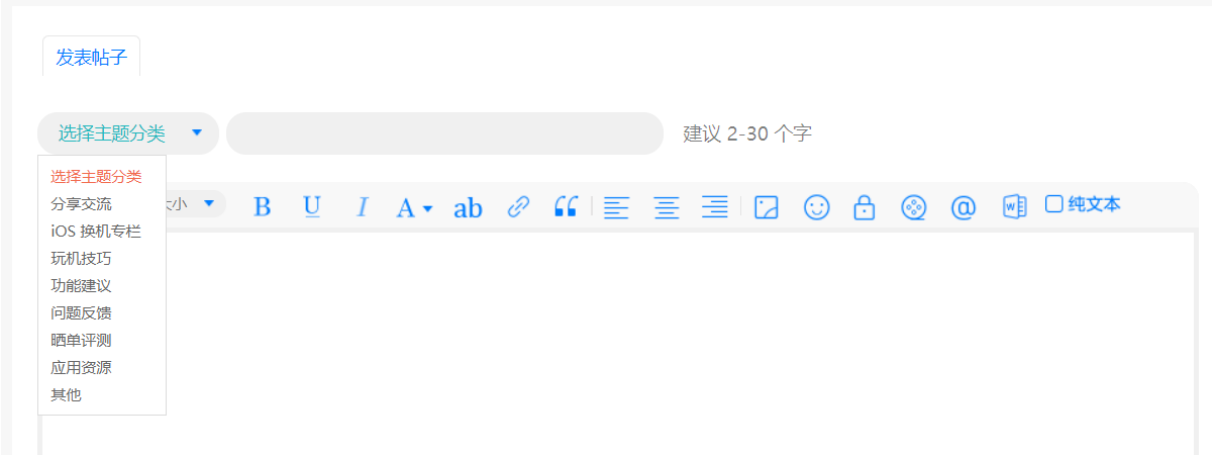


图 4-1 用户发布新帖子手动选择话题类型

我们在进行爬取花粉社区时，帖子所属的话题标签已被囊括。但这种让用户手动选择话题类型并不是没有差错的。事实上，用户在手动选择话题类型时存在与企业社区管理人员的认知偏差，比如说“玩机技巧”一栏，企社区管理人员希望用户在这一栏下分享自己使用其产品的技巧心得，比如说“冬天如何使用 Mate30 拍雪景更好看”、“相机 Pro 模式参数如何设置”等。但实际在社区运营过程中，部分用户却选择了这一话题类型用于表述自己在产品使用过程中遇到的问题，比如图 4-2 中的第 4 个用户帖子的话题分类应属于“分享交流”而非“玩机技巧”。除开对话题类型的概念的认知偏差，无可否认的是存在部分用户在选择话题类型时是没有耐心的，从而胡乱选择一个话题类型。如本章开头所言，本章的任务是进行文档粒度的用户创意观点挖掘。从普遍的角度来看，包含用户创意观点的帖子归类于两种话题类型，即问题反馈和功能建议。鉴于用户帖子的话题类型存在混乱的问题，本章提出了一种文档粒度的非用户创意观点二分类过滤模型。

图 4-2 部分用户帖子的话题类型选择不当

4.2 传统单嵌入层卷积神经网络

传统的单嵌入层卷积神经网络通常由四个子网络构成，如图 4-3 所示。其中，输入层完成输入的本文到词嵌入表示的映射；卷积层通常包括多种不同大小的卷积核，用于提取不同的文本局部特征图（类似于 N-gram）；经卷积后的数据送入最大池化层可以对数据进行再一次特征提取；这些特征最终经由全连接操作送入 Softmax 层完成各类别概率的输出。下面将对每层的具体结构和功能进行详细的阐释。

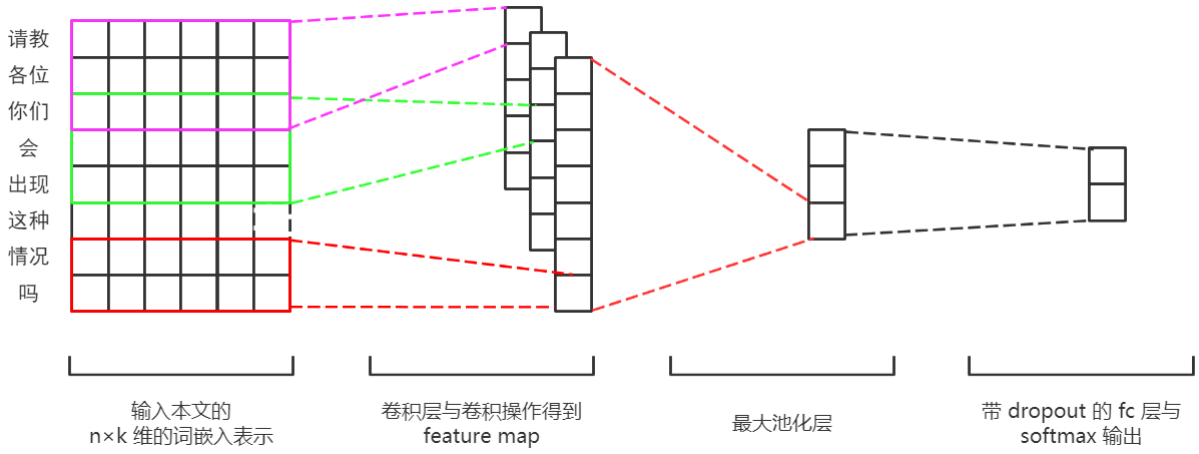


图 4-3 单通道卷积网络结构示意图

假设经分词后的输入文本的长度为 L，此处采取 padding = n 的策略，若 L > n，则只截取输入文本长度为 n 的前部分，若 L < n，则在输入文本后面补上 n – L 个“<PAD>”占位符以对齐（“<PAD>”存在对应的词嵌入）统一输入文本的长度后，对于输入文本的第 i 个词可以用一个 K 维的词嵌入向量表示，即 $X_i \in \mathbb{R}^K$，因此输入文本可以表示为：

$$\mathrm{X}_{1:n} = X_1 \oplus X_2 \oplus \cdots \oplus X_n \tag{4.1}$$

其中，$\oplus$ 是拼接符号，用于拼接两个向量，形成一个二维矩阵。$\mathrm{X}_{i:j}$ 则表示向量 $X_1, X_2, \cdots, X_n$ 的拼接。卷积运算则是通过一种称之为卷积核（也叫滤波器）的参数矩阵来进行，卷积核形如 $W \in \mathbb{R}^{HK}$。当卷积核作用于目标张量上，可产生新的特征图。公式为：

$$C_i = f(w \cdot X_{i+h-1} + b) \tag{4.2}$$

其中 f 是非线性激活函数，常用 Sigmoid、tanh 等函数，h 是卷积核滑动窗口宽度，b 是对应的偏置项。当卷积核在输入文本上滑动分别作用于 $\{X_{1:h}, X_{2:h+1}, \cdots, X_{n-h+1:n}\}$ 时，会产生一系列的特征图：

$$C = [C_1, C_2, \cdots, C_{n-h+1}] \tag{4.3}$$

然后我们对得到的特征图集合进行最大池化操作，即取 $\hat{C} = \max(C)$ 作为由该卷积核产生的特征图。以上是单个卷积核作用于输入文本的操作，在实际操作中，通常会设置不同滑动窗口大小的卷积核重复上述操作。最后将所有经由卷积、池化得到特征图拼接在一起，得到最大池化层。最大池化层的特征经全连接操作被传递给 softmax 层，并输出一个概率分布向量，其中，softmax 层的每个神经网络结点输出计算如下：

$$p(y_k | \hat{C}) = \frac{\exp(w_k * \hat{C} + b_k)}{\sum_{i=1}^{n} \exp(w_k * \hat{C} + b_k)} \tag{4.4}$$

其中 w_k 和 b_k 分别是全连接层与 softmax 输出层的连接权值与偏置值。假设类别标签数为 m，则上述模型的最终输出为向量 $\{p_1, p_2, \ldots, p_m\}$，并选取该向量的最大分量 p_i 对应的第 i 个类别作为输入文本 X 的最终类别。这里选择分类模型常用的交叉熵损失函数作为该网络模型的损失函数：

$$Loss = -\sum_{i=1}^{n} log\ p(y_k | x_i, \Theta) \tag{4.5}$$

Θ 是模型参数，通过反向传播不断计算逼近预测与实际类别标签的交叉熵误差的最小值，直到收敛或完成规定迭代次数。

在 CNN 模型训练过程中，如果训练样本不多，为了避免参数过多而导致的过拟合使得模型泛化性能不好，通常会在全连接层加入 dropout 机制。dropout 机制的原理是每次批量训练时会以一定概率丢弃某些网络连接，从而减少参数。一般 dropout 值设定为 0.5，即随机丢弃一半的参数。此外，也通常采用 L2 正则化来对卷积神经网络的参数进行约束，以尽量避免模型在训练中出现过拟合的现象。在本课题的模型试验中，皆采用梯度下降法完成最优模型参数的求解，样本迭代方式则采用最小批量梯度下降法（MBGD）。

4.3 多嵌入层 CNN 模型架构设计

在 4.2 节中介绍了传统的单嵌入层卷积神经网络（Single-Embedding CNN，SE-CNN），在这节中，将对该模型进行一定的改进，从而提出多嵌入层卷积神经网络（Multi-Embedding CNN，ME-CNN）。这里的多嵌入层由输入文本经过不同的预训练词向量模型映射得到。可以是基于跳字模型或连续池袋模型训练得到的词向量模型，也可以是随机生成的词向量模型等。因为这些不同的预训练词向量模型各有优点，因此能在词嵌入表示上加权，相互补充，进而能更有效提取输入文本的局部语义特征。后面的实验证明，SE-CNN 的效果优于各自的 SE-CNN。为了方便表示 ME-CNN 模型，这里不妨取嵌入层数为二，则 2E-CNN 模型的架构如图 4-4 所示。可以看到，和 SE-CNN 模型相似，2E-CNN 模型仍由四个子网络组成，其中最大的区别在于输入层：2E-CNN 模型中每个输入文本会被分别映射为各自的词嵌入表示，之后经过卷积、池化操作产生对应的特征图，并在最大池化层合并，最终经由全连接操作传递给 softmax 输出层。

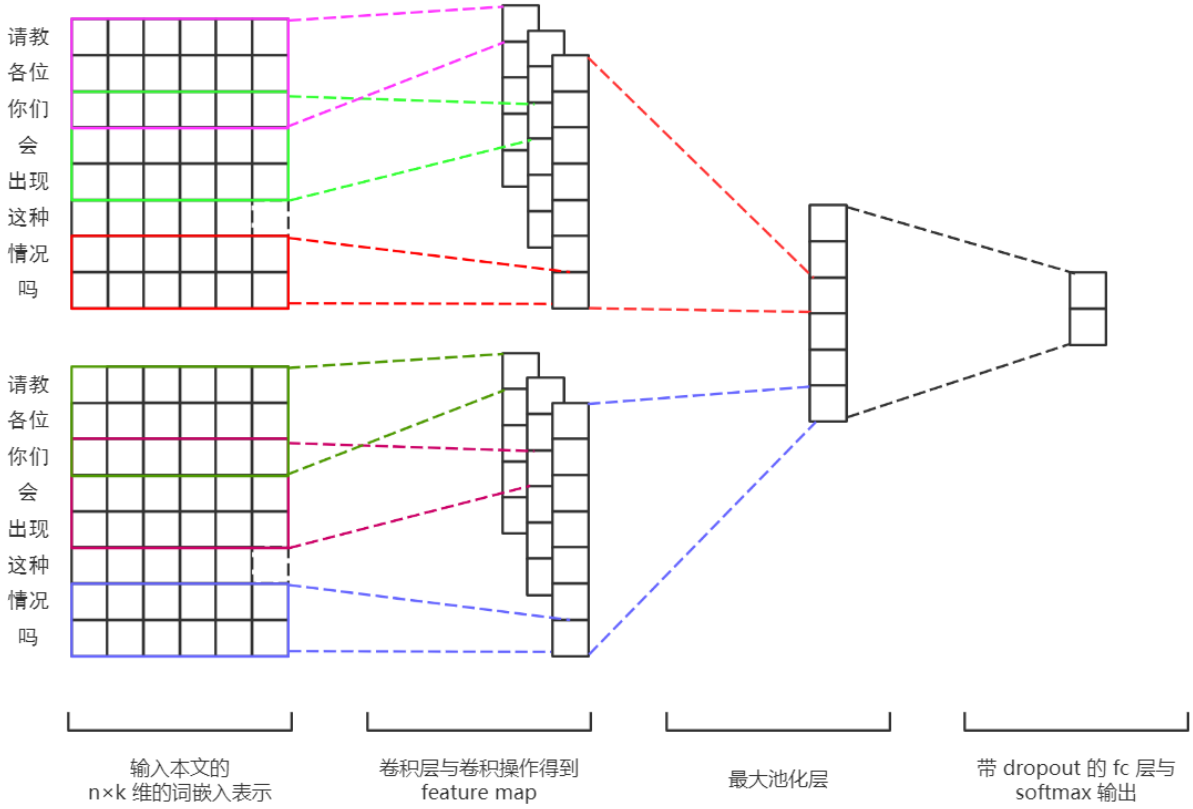


图 4-4 双嵌入层卷积网络结构示意图

2E-CNN 模型的计算公式与 4.2 节传统的单嵌入层 CNN 相同。值得注意的是，虽然计算过程相同，但分别作用于两个嵌入层各自卷积核的参数矩阵并不共享，也就是说他们是相互独立的。但为了保证后续最大池化操作的一致性，多个卷积核的滑动窗口的参数设置是相同的，在本章的实验中皆设定为（2, 3, 4）。另一方面，尽管增加了嵌入层的数量，从一定意义而言，这仅是增加了卷积 channel 的数量，因而在实际训练时进行的仍是一维卷积操作。这一点与图像卷积操作是有实质区别的。同时，尽管增加了额外的嵌入层会带来额外的计算开销，但因为对这两个嵌入层的分别的卷积、池化是分开的，直到最大池化层方才合并特征图。换而言之，相当于同时训练两个 SE-CNN 模型，因此，ME-CNN 模型的训练时间复杂度相对于 SE-CNN 而言，是线性的增加。

为了避免模型参数过多导致的过拟合现象， ME-CNN 模型采取和传统 SE-CNN 模型相同的 L2 范数正则化策略和在全连接层添加 dropout 机制。在这个基础上，ME-CNN 模型同时在嵌入层也加入了 dropout 机制。为了增加实验的可信性，在后面的 SE-CNN 模型的嵌入层中也加入 dropout 层，最优 dropout 值通过超参数实验得出，以求对标。最后，ME-CNN 模型训练的样本迭代方式与 SE-CNN 相同，皆使用 MBGD 策略。

4.3 模型实验

4.3.1 实验相关说明

（1）实验环境

如表 4-1 所示。

表 4-1 实验环境

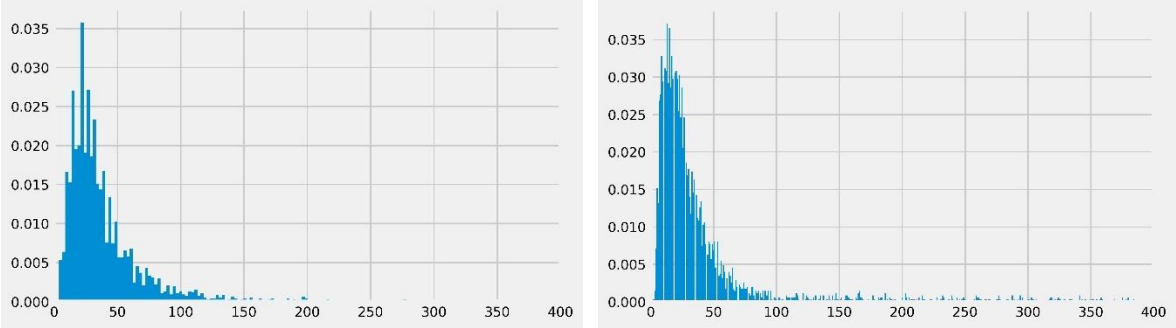
实验环境	具体配置
操作系统	Windows 10 Education 1909
CPU	Intel(R) Core(TM) i7-7700HQ CPU @2.80GHz
内存	16 GB
GPU	GeForce GTX1050 Ti
CUDA	CUDA 10.1.152
编程语言	Python 3.7
深度学习框架	Pytorch 1.4.0

（2）实验数据

本章实验数据来自第三章 3.2.2 节得到的主题数据集，采用 0.6: 0.2: 0.2 的比例对数据集进行划分，具体数值见表 4-2。同时，对于输入到模型的句子的长度设置也是重要的，图 4-5 的(a)和(b)分别展示了主题数据集中正负样本的句子长度的分布情况。可以看出数据集中大部分帖子长度处于 0 到 100，观察到这一点，模型对于输入句子的固定长度设置为 100，以避免数据稀疏问题。

表 4-2 主题数据集样本分布及划分

样本类型	训练集	验证集	测试集	合计
正样本	1200	400	400	2000
负样本	1200	400	400	2000
合计	2400	800	800	4000



(a) 正样本句子长度分布　　(b) 负样本句子长度分布

图 4-5 句子长度分布

（3）超参数设置

部分参数配置如表 4-3 所示。

表 4-3 超参数配置

参数说明	值
词嵌入维度	由实际而定
句长限制	100
批训练样本数	64
卷积核尺寸	(2, 3, 4)
卷积核数量	256
全连接层 dropout	0.5
优化器	Adam
迭代次数	10

（4）对比实验设置

本章主要通过控制嵌入层的类型和数量来进行对比实验。分别选取了第三章 3.3.3 节的 CBOW_300D、SG_300D 和 RAND_300D 等三个 300 维的词向量模型，并加以组合，得到以下 7 个模型，如表 4-4 所示。但要说明此处设置的各模型的词向量维度仅为示例，最佳取值由本章 4.3 节的超参数实验结果选出。

表 4-4 对比实验模型

模型名称	嵌入层说明
CNN-C	CBOW_300D
CNN-S	SG_300D
CNN-R	RAND_300D
CNN-CS	SG_300D + SG_300D
CNN-CR	CBOW_300D
CNN-SR	SG_300D + RAND_300D
CNN-CSR	CBOW_300D + SG_300D + RAND_300D

4.3.2 实验结果分析

本章实验分为两部分，第一部分探究 SE-CNN 和 ME-CNN 的最佳词向量维度和最佳词嵌入层 dropout 值。第二部分为验证本章提出的 ME-CNN 模型的可行性，而将 2E-CNN 模型、3E-CNN 模型和 SE-CNN 模型进行对标实验。

（1）超参数选择实验

在这次实验中，分别使用第三章 3.3 节训练得到的 SG_100D、SG_200D、SG_300D 等三个预训练词向量模型，分别在词嵌入层 dropout 取值为 0、0.1、0.2、0.3、0.4、0.5 下进行实验。同时为了简化实验，这里选取了表 4-4 中具有代表性的 CNN-C、CNN-CS、CNN-CSR 等三个模型参与实验，并以模型准确率作为评价指标。实验结果如表 4-5、表 4-6、表 4-7 所示。

表 4-5 CNN-C 实验结果

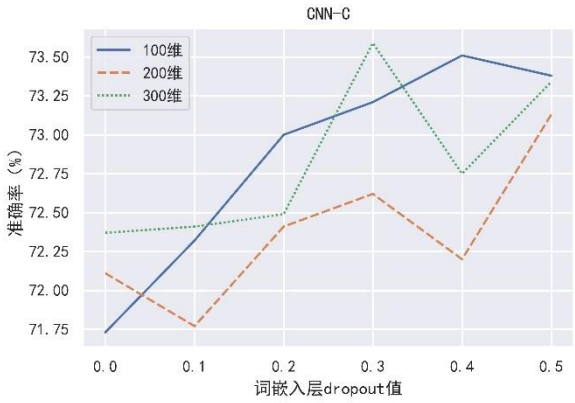
词向量维度	0	0.1	0.2	0.3	0.4	0.5
100 维	71.73	72.32	73.00	73.21	73.51	73.38
200 维	72.11	71.77	72.41	72.62	72.20	73.13
300 维	72.37	72.41	72.49	73.59	72.75	73.34

表 4-6 CNN-CS 实验结果

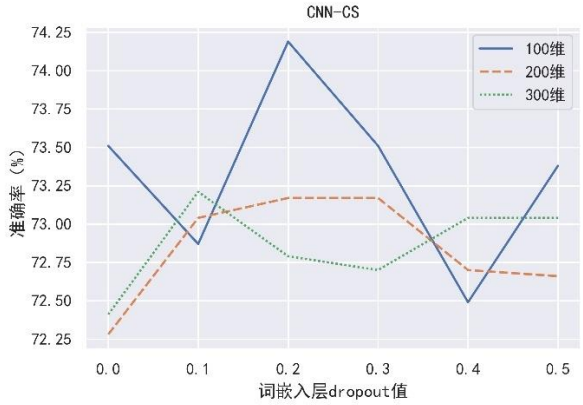
词向量维度	0	0.1	0.2	0.3	0.4	0.5
100 维	73.51	72.87	74.19	73.51	72.49	73.38
200 维	72.28	73.04	73.17	73.17	72.70	72.66
300 维	72.41	73.21	72.79	72.70	73.04	73.04

表 4-7 CNN-CSR 实验结果

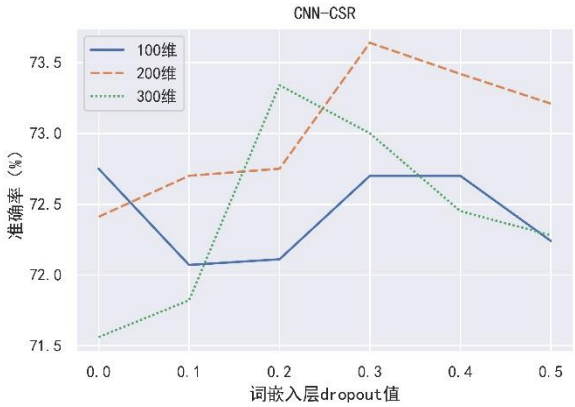
词向量维度	0	0.1	0.2	0.3	0.4	0.5
100 维	72.75	72.07	72.11	72.70	72.70	72.24
200 维	72.41	72.70	72.75	73.64	73.42	73.21
300 维	71.56	71.82	73.34	73.00	72.45	72.28



(a) CNN-C 实验结果



(b) CNN-CS 实验结果



(c) CNN-CSR 实验结果

图 4-6 不同嵌入层超参数实验结果

为了使实验结果更直观，这里采用可视化展示，如图 4-6 的所示。容易得出三类不同嵌入层模型的最佳参数配置。因此，在本章后续的实验中，对于单嵌入层 CNN，采用 300 维词向量维度，词嵌入 dropout 取值 0.3。对于双嵌入层 CNN，采用 100 维词向量维度，词嵌入 dropout 取值 0.2。对于三嵌入层 CNN，采用 200 维词向量维度，词嵌入 dropout 取值 0.3。同时，如图 4-6 所示，相比于传统的词嵌入层无 dropout 机制的卷积神经网络，实验表明往词嵌入层加入 dropout 机制在多数情况下有助于提升模型性能。

（2）ME-CNN 模型对比实验

本次实验各模型的词向量维度以及词嵌入层 dropout 值采用了本章实验（1）的最佳实验结果。其余超参数设置见表 4-3。将表 4-4 的 7 个模型分别作用于主题数据集，进行文本分类任务。以平均 F1 分值作为主要评价标准，固定超参数和数据集先后进行 3 次实验，统计实验结果取平均，实验结果如表 4-8 所示。

表 4-8 实验结果

模型	精确率（%）			召回率（%）			F1 值（%）		
	Neg	Pos	Avg	Neg	Pos	Avg	Neg	Pos	Avg
CNN-C	72.89	74.30	73.59	74.01	73.18	73.60	73.45	73.74	73.59
CNN-S	74.48	71.03	72.76	67.58	77.44	72.51	70.86	74.10	72.58
CNN-R	70.47	70.62	70.54	69.38	71.68	70.53	69.92	71.14	70.53
CNN-CS	74.09	74.28	74.18	73.33	75.02	74.17	73.71	74.65	74.18
CNN-CR	71.51	74.24	72.87	74.70	71.01	72.86	73.07	72.59	72.83
CNN-SR	72.16	71.11	71.63	69.13	74.02	71.57	70.61	72.53	71.60
CNN-CSR	73.03	74.24	73.63	73.84	73.43	73.64	73.43	73.83	73.63

首先，本章设计了三种词嵌入表示，分别是基于“花粉+维基语料”采用 CBOW 模型训练出来的 CBOW 词向量模型（简记为 C）、基于“花粉数语料”采用 Skip-gram 模型训练出来的 SG 词向量模型（简记为 S）以及随机生成的 Random 词向量模型（简记为 R）。对于单嵌入层 CNN，从表 4-8 可以看出，CNN-C 表现最佳，CNN-S 次之，CNN-R 较差，同时也说明了词向量模型是有效的。

从平均 F1 分值来看，CNN-CS 模型比 CNN-C 和 CNN-S 模型分别提升了 0.59%、1.6%，实验表明结合两种预训练词向量模型的 CNN-CS 的两个嵌入层能够互为补充，各展所长。但反观 CNN-CR 和 CNN-SR 两个双嵌入层模型的 F1 分值相比 CNN-C 和 CNN-S 两个单嵌入层模型要分别降低 0.76%、0.98%，以及 CNN-CSR 三嵌入层模型的 F1 分值要比 CNN-CS 模型下降 0.55%，这两点实验数据都能说明为卷积神经网络增加随机生成的嵌入层会对模型性能的提升起反效果。

4.4 本章小结

本章首先简单介绍了用户帖子主题分类的问题，然后重点展开了传统卷积神经网络的架构以及原理，并在此基础上提出改进：多嵌入层表示，并在嵌入层加入 dropout 机制。为了验证模型改进的有效性，并找到最适合的词向量维度和嵌入层 dropout 值，进行了两部分实验。第一部分为超参数选择实验，第二部分为验证对比实验。实验结果表明，增加预训练词向量模型的嵌入层可以提升卷积神经网络在文本分类任务中的性能，增加随机生成的嵌入层则会起反效果。当词向量维度选择 100 维，词嵌入 dropout 取值 0.2，双嵌入层 CNN 模型能取得最好性能；当词向量维度选择 200 维，词嵌入 dropout 取值 0.3，三嵌入层 CNN 模型能取得最好性能。

第五章　基于组合模型的非创意观点信息过滤

5.1 问题描述

上一章完成了文档粒度的不含有用户创意观点的用户帖子的主题过滤，这一章将进行更小粒度——句子粒度——的非用户创意观点的句子过滤，尝试在包含用户创意观点的用户帖子中识别出真正包含用户创意观点的句子。尽管 2014 年 Kim[20]提出的 TextCNN 中的卷积层能够有效捕捉输入文本的局部语义信息，但在句子粒度的观点过滤任务中，却存在着局限性。考虑以下两个句子："Mate30 的夜间拍摄模式真的太棒了！"和"Mate30 的夜间拍摄模式一点都不好用！"。TextCNN 能够对这两个句子中的局部特征"夜间拍摄模式"进行捕捉，但问题是，第一个句子属于非用户创意信息句子，第二个句子则属于用户创意信息句子。因此，在句子粒度的观点过滤任务中，还需要考虑语义的前后关系，比如说，第一个句子中要考虑"夜间拍摄模式"后面的"太棒了"，第二个句子中要考虑"夜间拍摄模式"后面的"不好用"。TextCNN 无法对这种依赖关系进行捕捉的。

为了捕捉这种前后的依赖关系，通常的做法是使用 RNN 模型。RNN 模型由于其结构特性，可以天然加入语序信息，同时对前后词之间的依赖建模。或者使用改进的 BiLSTM 进行建模。关于 RNN 和 BiLSTM 的具体信息，在第二章已进行综述，这里不再展开。但值得指出的是，RNN 和 BiLSTM 模型的串行结构特点是其处理序列数据的天然优势所在，但也是其天生的缺陷来源，即无法将数据交由图形处理单元（GPU）进行并行化计算，极大幅度限制了模型训练的时间性能的减少。随后在 2017 年，Google 团队提出 Transformer 模型，其与 BiLSTM 一样，皆属于 Seq2Seq 模型，并都能进行长距离依赖建模；区别在于 Transformer 能进行并行化计算，而 BiLSTM 不能。Transformer 的出现消弭了传统序列模型不能进行并行化计算的缺陷。本章将结合 Transformer 能够进行长距离依赖建模、可并行化处理和 CNN 能够抽取局部语义特征的特点，提出 Transformer 和 CNN 的组合模型（为了表述方便，下文简称之为 TF-CNN 模型）。

5.2 TF-CNN 模型架构设计

本章提出的 TF-CNN 模型架构如图 5-1 所示。模型整体由两个子网络构成，第一个子网络为 Transformer 的编码器模块，由若干个编码器串联得到，第二个子网络则是第四章 4.2.1 节中的单嵌入层卷积神经网络。下面主要介绍 Transformer 的编码器模块的结构及原理。

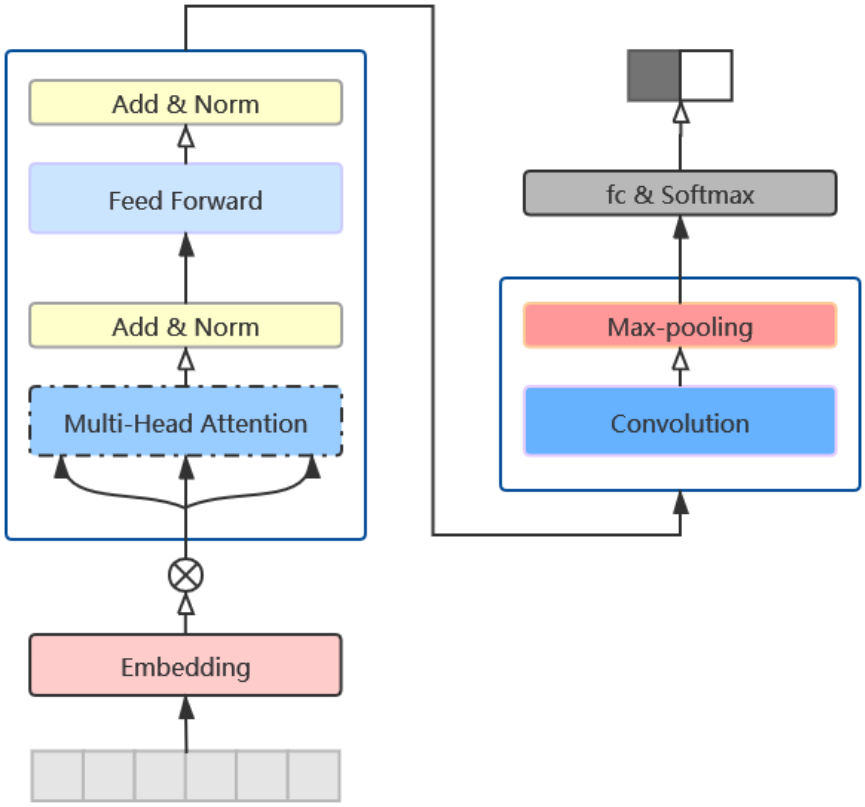


图 5-1 TF-CNN 模型架构

每个编码器都包含两个组件：多头注意力网络和前馈神经网络。其中，多头注意力网络的功能是为输入到当前编码器的词嵌入矩阵的每一个词向量分别生成一个上下文向量，上下文向量包含原词向量的上下文语境信息。前馈神经网络的功能则是将嵌入矩阵的每个原词向量和对应新生成的上下文向量的进行整合，从而生成了考虑整个句子（词嵌入矩阵）上下文的当前时刻的隐含状态。

首先，输入序列经过词嵌入层，得到词嵌入矩阵 $x = (x_1, x_2, \ldots, x_n)$，并加入位置编码，生成包含词序信息的词嵌入表示。位置编码的计算公式为：

$$PE_{pos,2i} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \tag{5.1}$$

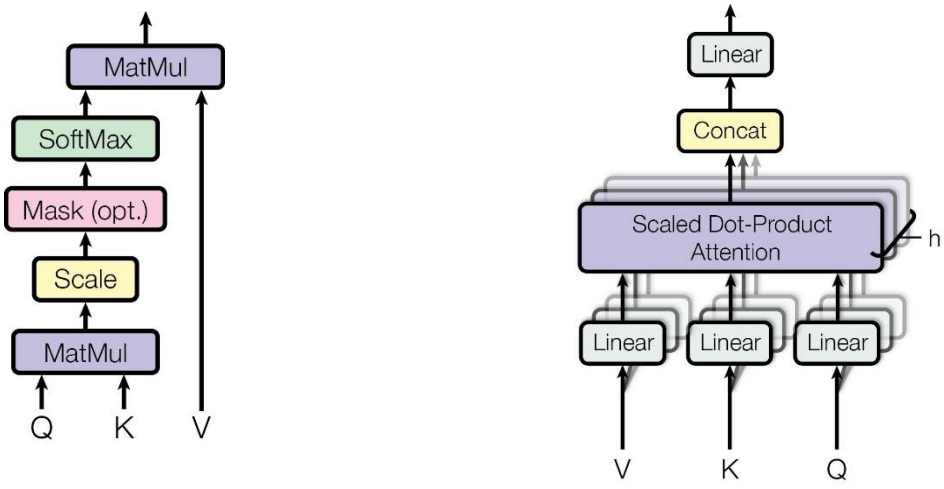
$$PE_{pos,2i+1} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \tag{5.2}$$

pos 是该词在输入文本中的位置，d 是词向量维度，i 是词向量的第 i 个维度。上述过程可以简记为 $a = (a_1, a_2, \dots, a_n) = W \cdot x$。分别对 a_i 进行三次线性变换，即 $Q = W^Q \cdot a_i$，$K = W^K \cdot a_i$，$V = W^V \cdot a_i$，则 a_i 进入自注意力层后的输出为：

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V \tag{5.3}$$

其中，Q 表示查询向量，用于匹配其他向量；K 表示键向量，用于配合 Q 匹配其他向量，V 表示值向量，表示需要被抽取的信息。单头计算过程 $head_i = Attention(Q, K, V)$，如图 5-2 (a)所示。当所有头计算完毕，如图 5-2 (b)所示，拼接计算结果，并将拼接结果做一次线性变换，即：

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h) \cdot W^O \tag{5.4}$$



(a) 单头注意力计算　　(b) 多头注意力计算

图 5-2 注意力结构[33]

为了避免数据分布不一致，随后对多头注意力网络的输出做了一次 Layer Normalization，即图 5-1 中的 Add & Norm 层。最终数据进入前馈神经网络：

$$FFN(x) = \max(0, x \cdot W_1 + b_1) \cdot W_2 + b_2 \tag{5.5}$$

5.3 模型实验

5.3.1 实验相关说明

（1）实验环境

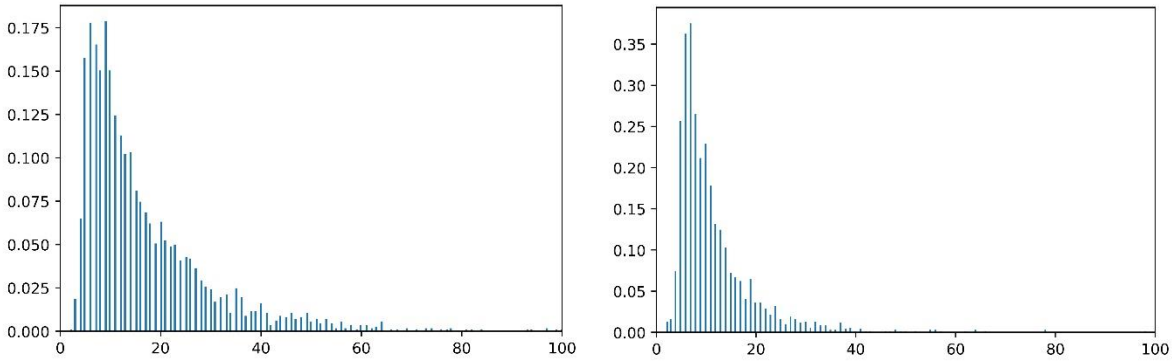
本章实验环境同第四章，见表 4-1。

（2）实验数据

本章实验数据来自第三章 3.2.3 节得到的观点数据集，采用 0.6: 0.2: 0.2 的比例对数据集进行划分，详细数目见表 5-1。同时，对于输入到模型的句子的长度设置也是重要的，图 5-3 的(a)和(b)分别展示了主题数据集中正负样本的句子长度的分布情况。可以看出数据集中大部分帖子长度处于 0 到 60，观察到这一点，模型对于输入句子的固定长度设置为 60，以避免数据稀疏问题。

表 5-1 主题数据集样本分布及划分

样本类型	训练集	验证集	测试集	合计
正样本	1240	330	330	1900
负样本	800	220	221	1241
合计	2040	550	551	3141



(a) 正样本句子长度分布　　(b) 负样本句子长度分布

图 5-3 正样本句子长度分布

（3）超参数设置

如表 5-2 所示。

表 5-2 超参数设置

参数说明	值
词嵌入维度	300
句长限制	60
批训练样本数	64
模型维度	300
隐藏层	1024
最后隐藏层	512
注意力头数	5
编码器数量	2
卷积核尺寸	(2, 3, 4)
卷积核数量	256
全连接 dropout	0.5
优化器	Adam
迭代次数	20

（4）基准实验设置

为检验本章提出的 TF-CNN 模型在用户创意观点过滤任务中的可行性，本章在以下六个基准模型上进行对标实验，见表 5-3。

表 5-3 基准模型

模型名称	说明
TextCNN	由 Kim[20]在 2014 年提出，首次将 CNN 应用于文本分类任务。
TextRNN	由 Pengfei Liu 等人[21]在 2016 年提出，作者针对单任务训练数据不足的问题，提出了多任务学习模型。该模型基于递归神经网络完成数据传递，并取得了不错的分类效果。
BiLSTM_Att	由 Zhou[35]在 2016 年提出，作者使用了基于注意力的双向长短期记忆网络来捕获句子中最重要的语义信息，用于关系分类。
RCNN	由 Lai[36]在 2015 年提出，作者设计了一种比传统的基于窗口的神经网络噪声要小的双向循环结构，能够最大化地提取上下文信息。
DPCNN	由 Johnson 和 Zhang[37]提出，作者提出了一种单词级别的深层 CNN 模型，通过不断加深网络，来捕捉文本的全局语义表征。
Transformer	由 Google 团队在 2017 年提出[33]。

5.3.2 实验结果与分析

本章实验分为两部分进行，首先为 TF-CNN 模型的进行超参数选择实验，随后将 TF-CNN 模型和六个基准模型进行对比实验，以验证 TF-CNN 模型的可行性。

（1）超参数选择实验

相对而言，对 TF-CNN 模型影响较大的超参数为：编码器模块的注意力头数以及编码器数量。为考察这两类参数对模型效果、训练时间的影响，以及找到适合本章用户创意观点过滤任务的较佳参数，接下来将分别进行两组超参数选择实验。

a. 注意力头数选择实验

表 5-4 注意力头数选择实验结果

num_head	1	2	3	4	5	6	7	8	9	10	11	12
准确率（%）	76.70	78.10	77.70	77.40	78.50	77.30	-	-	-	77.90	-	76.40
训练时间	02:50	02:53	02:54	02:54	02:56	02:57	-	-	-	03:04	-	03:09

注：训练时间的格式为“分钟：秒钟”

因为模型维度设值为 300，其必须整除注意力头数，所以表 5-4 部分实验项没有数值。在本实验中，分别选取了 1-12 的注意力头数，从表 5-4 可以看出，注意力头数因为由 GPU 进行并行化计算，所以对模型训练时间影响有限。而最佳的实验结果的注意力头数取值为 5，此时模型准确率为 78.5%。因此，本章实验中的 TF-CNN 模型的注意力头数宜取 5。

b. 编码器数量选择实验

表 5-5 编码器数量选择实验结果

num_encoder	1	2	3	4	5	6	7	8	9
准确率（%）	78.10	78.50	73.70	75.60	75.80	58.74	58.74	58.74	58.74
训练时间	02:03	02:54	03:46	04:36	05:30	06:22	07:13	08:02	08:56

注：训练时间的格式为“分钟：秒钟”

在本实验中，分别设置了 1-9 的编码器数量，如表 5-5 所示，可以看到训练时间随编码数量的增加而线性增长，这是因为编码器在 TF-CNN 模型内部是以串联的方式连接。模型的准确率则是先随着编码器数量的增加而得到提升，但在编码器数量为 2 的时候达到的峰值，并当编码器数量超过 5 后，模型性能出现断崖式的下跌，并且准确率不再发生变化，此时模型参数过多，从而发生了过拟合的情况。综上分析，编码器数量宜取 2。

（2）基准模型对比实验

将 TF-CNN 和六个基准模型分别作用于观点数据集，进行用户创意观点过滤任务，以精确率、召回率和 F1 值作为模型评价标准，但因为正负样本各自的精确率、召回率不能一概而论，如果简单进行平均也是不合理的，因此主要以精确率和召回率的调和平均数 F1 分值作为主要评价标准。固定超参数和数据集分别进行 3 次实验，并对实验结果进行平均，实验结果如表 5-6 所示。

由表 5-6 可以看出，从 F1 值来看，本章提出 TF-CNN 模型（即 Transformer、CNN 的组合模型）要分别比 Transformer、CNN 提高 3.15%、1.41%，同时也比基准模型中表现最好 RCNN 模型要高出 0.38%。从 TF-CNN 模型整体来看，Transformer 的编码器模块发挥的作用类似于一个语义特征提取器，而 CNN 对 Transformer 的特征提取结果进一步提取。TF-CNN 模型的分类性能要优于单独的 Transformer 或 CNN 模型，表明 TF-CNN

模型可以有效结合 Transformer 能够进行长距离依赖建模从而加入词之间相似结构和 CNN 能够捕捉文本序列的局部语义信息的优点，从而提升分类模型的整体性能。

表 5-6 实验结果

模型	精确率（%）			召回率（%）			F1 值（%）		
	Neg	Pos	Avg	Neg	Pos	Avg	Neg	Pos	Avg
TF-CNN	72.96	82.66	77.81	75.97	80.27	78.12	75.54	81.45	78.50
TextCNN	76.36	79.27	77.82	68.20	85.20	7670	72.05	82.13	77.09
TextRNN	70.56	77.89	74.22	67.48	80.27	73.87	68.98	79.06	74.02
BiLSTM-Att	78.89	78.30	78.59	65.29	87.76	76.52	71.45	82.76	77.10
RCNN	76.03	80.88	78.46	7160	84.18	77.89	73.75	82.50	78.12
DPCNN	64.64	84.81	74.72	82.52	68.37	75.45	72.49	75.71	74.10
Transformer	71.75	79.17	75.46	69.66	80.78	75.22	70.69	79.97	75.33

5.4 本章小结

本章首先从用户创意观点过滤任务出发，介绍了句子粒度的文本分类的常见模型，同时指出了这些模型的不足，进而提出 TF-CNN 模型。随后重点介绍了 TF-CNN 的整体构架和 Transformer 的 Encoder 模块的原理。本章最后进行了两部分的实验，第一部分为超参数优化实验，分别设置了注意力头数、编码器数量两组超参数，通过实验选取最佳参数值；第二部分为 TF-CNN 模型和六个基准模型的对比实验。实验结果表明，TF-CNN 模型可以有效结合 Transformer 和 CNN 的优点，发掘更丰富的语义信息，进一步提高用户创意观点过滤模型的性能。

第六章　基于层次凝聚聚类的用户创意观点聚类

6.1 问题描述

本章不是本课题的研究重点，只进行一些试探性工作。经过第四章的主题分类以及第五章的用户创意观点过滤，可以得到较为纯粹的用户创意观点信息。考虑到在第五章的实验结果中，可能存在较多的相似度高甚至重复冗余的用户创意观点信息，比如说，用户 A 说：“自己看吧 30 分钟待机掉了百分之 4 的电！”，用户 B 说：“用电真的好快待机 12 个小时就不见百分之 25 电了”，用户 C 说：“我这一晚上待机 9 小时掉了百分之八的电。”。尽管这些用户表述观点信息的遣词造句、用语习惯等形式不同，但本质上表达的内容是一样的，即手机待机耗电快。这些表达形式不同，但表达内容实质一样或高度相似的用户创意观点信息应该被归为一类。又因为数据属于无标签数据，同时类别数是不确定的，因此数据挖掘领域的聚类算法比较适合本章的业务场景。

聚类分析中有两种代表性的数据结构作为聚类对象，一种是 $n \times m$ 规模的数据矩阵，即共有 n 个待聚类的对象，分别用 m 个属性来表示一个对象。另一种则是 $n \times n$ 规模的相异度矩阵，分别存储 n 个对象两两之间的差异度。这里选择了第一种数据结构，这也就意味着每一个参与聚类的对象（用户创意观点）都是 $1 \times m$ 大小的向量，而在第五章中，每个句子被映射为 $k \times m$ 的矩阵，对此，我们需要再进行一次映射。常见的做法是使用均值模型生成句向量，即 $sen = \frac{1}{m} \sum_{i=1}^{k} v_i$，其中 v_i 是词向量。

6.2 基于 Ward Linkage 的聚类算法设计

本章使用基于 Ward Linkage 距离度量方法的分层凝聚聚类算法（Hierarchical Agglomerative Clustering，HAC），HAC[40]采用自底向上的操作，其主要思想是，首先初始化每个样本点为一个独立的簇，通过度量簇间距离，每次合并距离最近的两个簇，迭代这个过程，直到满足迭代终止条件。这里给出 HAC 算法的主要步骤，见表 6-1。

表 6-1 HAC 算法步骤

输入：N 个数据点，目标簇数 K
输出：K 个簇（$\boldsymbol{1 \leq K < N}$）
1）初始化输入数据集中的每个数据点为一个簇；
2）分别计算每两簇之间的最小距离；
3）将两簇距离最小的两个簇合并为一个新簇；
4）重复 2）、3），直到得到的当前聚类数达到输入的目标簇数 K。

这里再简洁介绍 Ward Linkage，Ward Linkage 的核心概念是簇内所有数据点的离差平方和（Error Sum Of Squares，ESS）：

$$ESS = \sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2 \tag{6.1}$$

则基于 Ward Linkage 两簇距离度量方法的分层凝聚聚类算法在表 6-1 的第 3）步可表述为每次两个簇使合并后的新簇内的离差平方和最小。

6.3 模型实验

6.3.1 实验相关说明

（1）实验环境

如表 6-2 所示。

表 6-2 实验环境

实验环境	具体配置
操作系统	Windows 10 Education 1909
CPU	Intel(R) Core(TM) i7-7700HQ CPU @2.80GHz
内存	16 GB
GPU	GeForce GTX1050 Ti
编程语言	Python 3.7
聚类工具	Scikit-learn 0.22.1

（2）实验数据

本章实验数据来自第五章的实验结果，即包含用户创意观点的句子集合。

6.3.2 实验结果分析

经过实验，发现对于输入模型的数据，当聚类簇数取值 500 时，聚类效果较为理想。部分实验结果如表 6-3 所示。

表 6-3 聚类部分实验结果

```
"1": [
    "请求灭屏显示加入自定义文字内容",
    "希望增加简体转繁体功能。",
    "目前我知道的就只能备忘录加密码锁，锁定所有备忘录。",
    "备忘录待办设定了，到时间了没提醒。",
    "希望备忘录添加一个置顶功能。",
    "希望备忘录的待办事项能设置重复提醒",
    "联系人智能群组里单位能否自定义修改。"
],
"2": [
    "外放高音没有立体感，音质变差。",
    "外放音量在 60%以上破音。"
],
…
"78": [
    "mate30pro 屏幕是用的那家的？",
    "mate30 和 mate30pro 屏幕通病。",
    "mate30pro 屏幕按压有声响。",
    "华为 mate30pro 屏幕横纹。",
 ],
 …
"145": [
    "开启机械键盘，打字时振动有异常",
    "有时振动反馈异常振动时整只手机振动与仿真键盘震感不同",
    "只能水平振动，不能垂直振动？",
    "果然振动有问题。",
    "183 不是说修复振动反馈吗？",
    "195 振动又出现以前出现过的振动震感问题了！",
    "增加振动反馈的功能。"
 ],
 …
```

注：聚类结果使用键值对的方式表示，即“簇序号：簇”

本章对于聚类的实验结果并没有采取评价指标，而是选择进行人为评估。从表 6-3 可以看出，基于 Ward Linkage 的层次凝聚聚类算法可以依据一个或多个的关键词向量对实验对象进行聚类分析，如表 6-3 给出的第 1、2、78、145 个簇的关键词分别为“备忘录”，“外放”，“屏幕”，“振动”。尽管会出现一些“不速之客”，比如第 1 个簇中出现了“希望增加简体转繁体功能。”、“联系人智能群组里单位能否自定义修改。”等与“备忘录”无关的观点句子，但从实验总体来看，聚类效果还算不错。从另一个角度来看，层次凝聚聚类属于自底向上的聚类算法，算法无法在聚类簇数最合适时停下来，需要人工调参进行实验探索，这可能也正是大多数聚类算法的通病。但无论如何，由机器进行的聚类结果对人类而言是具有参考和启发价值的，它从一定程度上减轻重复性的工作，辅助人类更有效地进行思考决策。

6.4 本章小结

本章对用户创意观点聚类任务进行一些试探性工作。首先使用了均值模型生成句嵌入，随后介绍了基于 Ward Linkage 的层次凝聚聚类算法的原理和具体步骤。实验结果表明基于 Ward Linkage 的层次凝聚聚类算法适用于句子粒度的文本信息，并能取得不错的聚类效果。但同时模型也存在无法自动确定最佳聚类簇数的问题，有待进一步探索。

第七章　总结与展望

7.1 总结

传统的企业开放社区中的用户创意获取主要依赖于人工的方式。本课题基于深度学习技术，在前人文本表示和文本分类研究成果的基础上，针对企业开放社区的用户帖子进行主题分类、非创意观点信息过滤等两项主要工作，并对过滤后的句子进行试探性的观点聚类工作。完成的主要工作和结论如下：

（1）针对不同类型的词向量模型各有所长的特点，设计了 ME-CNN，并在传统 SE-CNN 的基础上增加了词嵌入 dropout 层。实验结果表明带有词嵌入 dropout 层多嵌入层的设计能增强输入文本的语义表示，其中 CNN-CS 模型分别比 CNN-C 和 CNN-S 模型提升了 0.59%、1.6%。

（2）为了捕捉长距离的语序信息和局部的语义信息，对深度学习技术的常用模型进行研究，提出 Transformer-CNN 混合模型，最后在观点数据集上完成了测试，F1 分值达到了 78.50%。并与 CNN、RCNN、Transformer 等 6 个基准模型的结果做对比分析，实验结果表明 Transformer-CNN 混合模型可以提取更深的文本特征，捕捉更丰富的语义信息，文本的分类性能得到了提高。

（3）对于大量的可能存在相似语义信息的观点集合，采取了基于 Ward Linkage 的层次凝聚聚类算法，对用户创意观点进行了聚类。结果人为评估，实验结果表明由均值模型生成的句嵌入适用于聚类分析，并能取得不错的聚类效果。

本文丰富了基于深度学习的企业开放社区中的用户创意挖掘方面的研究，对网络评论中的观点挖掘相关工作有一定参考意义。但就任务本身而言，尽管本课题提出的模型算法取得了不错的进展，但机器在这里是无法彻底替代企业员工的思考，因为只有企业员工本身才更了解他们做出来的产品。对于企业而言，人力资源是宝贵的，他们更适合在高层次上的问题进行思考，从而创造更高的价值。而对于主题分类、观点过滤、观点聚类等相对简单但耗费精力的重复性工作而言，交给机器，或许是极具智慧的选择，更符合企业的战略需求。机器对数据进行过滤、组织、分析，以辅助人类更有效地进行思考决策，甚至代替人类决策，或许这正是深度学习的研究意义，也是本课题的意义所在。

7.2 展望

在技术日新月异的互联网时代，信息也在指数级地爆炸增长，这对自然语言处理技术提出了更高的需求。本课题所尝试的针对企业开放社区的用户创意观点信心的挖掘技仅仅是自然语言处理领域中的众多子问题之一。尽管取得了一定成果，但仍然存在许多不足，需要进行下一步的研究：

（1）如今预训练模型大行其道，Word2vec 模型尽管在本课题中取得了不错的成果，但其本身却有着不可回避的缺陷性：一方面，Word2vec 属于静态词向量模型，在训练过程中无法进行微调（fine-tuning），无法针对特定任务做动态优化；另一方面，Word2vec 对于一词多义的问题无能为力，因为其词与词向量属于唯一对应关系。在 Word2vec 之后，有不少优秀的预训练模型被提出，如 ELMo[38]、Bert[39]等。以 Bert 为例，其克服了 Word2vec 无法表示多义词的缺点，同时 Bert 能表现词的多层特性，包括语义、语法等层面，具有诸多优点。因此下一步可以将 Bert 等新一代预训练模型引入到课题的模型中。

（2）在第四章和第五章的具体实验中，由于时间有限仅对其中部分重要超参数进行优化实验，因此在未来可以进行进一步分析和设计实验，进一步提升模型性能。

（3）在第六章中的实验中，面向句嵌入的聚类算法取得了不错的实验结果，但存在簇中出现“不速之客”的情况，因此有待进一步优化算法。

参考文献

[1] 原欣伟，王超超，杨少华. 开放式创新社区环境下领先用户参与创新研究综述 [J]. 未来与发展，2017, 041 (012):74-79.

[2] Hatzivassiloglou V, McKeown K R. Predicting the semantic orientation of adjectives[C]//Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics. Association for Computational Linguistics, 1997: 174-181.

[3] 王辉，王晖昱，左万利. 观点挖掘综述 [J]. 计算机应用研究，2009 (01):31-35.

[4] Valakunde N D, Patwardhan M S. Multi-aspect and multi-class based document sentiment analysis of educational data catering accreditation process[C]//2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies. IEEE, 2013: 188-192.

[5] Agarwal S, Yu H. Automatically classifying sentences in full-text biomedical articles into Introduction, Methods, Results and Discussion[J]. Bioinformatics, 2009, 25(23): 3174-3180.

[6] Brody S, Elhadad N. An unsupervised aspect-sentiment model for online reviews[C]//Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics. Association for Computational Linguistics, 2010: 804-812.

[7] 韩忠明，李梦琪，刘雯，等. 网络评论方面级观点挖掘方法研究综述 [J]. 软件学报，2018, 029 (002):417-441.

[8] 张茜，张士兵，任福继，等. 基于主题模型的微博评论方面观点褒贬态度挖掘 [J]. 中文信息学报，2019, 33 (6).

[9] BAEZA-YATES R, RIBEIRO-NETO B. Modern information retrieval [M]. ACM press New York,1999.

[10] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. Journal of machine learning research, 2003, 3(Feb): 1137-1155.

[11] ERK K, PAD S.A structured vector space model for word meaning in context[C]/Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics,2008:897-906.

[12] MAAS A L, DALY R E, PHAM P T, et al. Learning word vectors for sentiment analysis[C]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics,2011:142-150.

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

[14] 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.

[15] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.

[16] Mahmud M N, Ibrahim M N, Osman M K, et al.A robust transmission line fault classification scheme using class-dependent feature and 2-Tier multilayer perceptron network[J]. Electrical Engineering,2017:1-17.

[17] Kwon O W, Lee J H. Web page classification based on k-nearest neighbor approach[C]//Proceedings of the fifth international workshop on on Information retrieval with Asian languages. 2000: 9-15.

[18] McCallum A, Nigam K. A comparison of event models for naive bayes text classification[C]//AAAI-98 workshop on learning for text categorization. 1998, 752(1): 41-48.

[19] Joachims T. Svmlight: Support vector machine[J]. SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund, 1999, 19(4).

[20] Kim Y. Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv,2014.

[21] Liu P, Qiu X, Huang X. Recurrent neural network for text classification with multi-task learning[J]. arXiv preprint arXiv:1605.05101, 2016.

[22] Yang Z, Yang D, Dyer C, et al. Hierarchical attention networks for document classification[C]//Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. 2016: 1480-1489.

[23] Hubel D H, Wiesel T N. Receptive fields and functional architecture of monkey striate cortex[J]. Journal of Physiology,1968,195(1):215-243.

[24] Fukushima K, Miyake S, Ito T. Neocognitron: A neural network model for a mechanism of visual pattern recognition[J]. IEEE transactions on systems, man, and cybernetics, 1983 (5): 826-834.

[25] Khan A, Sohail A, Zahoora U, et al. A survey of the recent architectures of deep convolutional neural networks[J]. arXiv preprint arXiv:1901.06032, 2019.

[26] Potluri S, Fasih A, Vutukuru L K, et al. CNN based high performance computing for real time image processing on GPU[C]//Proceedings of the Joint INDS'11 & ISTET'11. IEEE, 2011: 1-7.

[27] Krizhevsky A, Sutskever I, Hinton G. ImageNet Classification with Deep Convolutional Neural Networks[C]//NIPS. Curran Associates Inc.2012.

[28] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model[C]. Eleventh annual conference of the international speech communication association, 2010.

[29] Hochreiter S , Schmidhuber J . Long Short-Term Memory[J]. Neural computation, 1997, 9(8):1735-1780.

[30] Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks[J]. arXiv preprint arXiv:1503.00075, 2015.

[31] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.

[32] Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.

[33] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.

[34]中国信通院．2019 年 12 月国内手机市场运行分析报告[R]．报告地：中国信通院，2019

[35]Zhou P, Shi W, Tian J, et al. Attention-based bidirectional long short-term memory networks for relation classification[C]//Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers). 2016: 207-212.

[36]Lai S, Xu L, Liu K, et al. Recurrent convolutional neural networks for text classification[C]//Twenty-ninth AAAI conference on artificial intelligence. 2015.

[37]Johnson R, Zhang T. Deep pyramid convolutional neural networks for text categorization[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017: 562-570.

[38]Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[J]. arXiv preprint arXiv:1802.05365, 2018.

[39]Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[40]Ward Jr J H. Hierarchical grouping to optimize an objective function[J]. Journal of the American statistical association, 1963, 58(301): 236-244.

致谢

光阴荏苒，岁月如梭，在广工大度过的四年本科生生涯即将要划上句号。此时此刻，内心感触良多，感谢在校园内度过的那些时光，感谢在这过程遇到的老师们、路上良友、室友以及班级同学。

首先，感谢我的论文指导老师唐洪婷老师。感谢她在我困惑不前的时候给予我解答与鼓励，感谢她在我论文写作过程中孜孜不倦、耳提面命地给予方向与修改意见。没有这些我将无法完成论文的选题、开题、实验到最终完成。唐洪婷老师对我影响最深的莫过于其敬业的精神和严谨的治学态度，这些让我一生受益。

然后，感谢共处四年的室友、朋友、同学，感谢他们在学习与生活中给予我无私的陪伴与帮助，很高兴与你们度过这些难忘的日子。

同时，我要感谢父母一直以来对我无微不至的照顾和关怀，他们始终愿意做我最坚强的后盾，让我无后顾之忧。他们的鼓励就是我求学路上最大的动力。

最后，衷心地感谢各位评审老师，感谢您们在百忙之中抽出时间来审阅论文，更感谢各位老师对论文提出的珍贵意见。

再次感恩所有帮助与关爱我的老师、朋友、同学、亲人们，感谢你们！