摘　要

随着互联网的高速发展，与日常生活息息相关的餐饮消费领域产生了大量在线评论文本，而针对评论文本的情感分析也随着相关理论的发展由粗粒度情感分析向方面级的细粒度情感分析延伸。本文采用“粗粒度评价维度识别+细粒度评价维度情感极性分类”的两阶段模式，基于深度学习方法完成餐饮消费在线评论文本的方面级情感分析任务。

本文以餐饮消费评论数据集 ASAP 作为研究对象，在第一阶段的粗粒度评价维度识别任务中，选取 LSAN 模型作为基准模型，针对其在特征提取层仅采用 Bi-LSTM 网络的缺陷提出 LSAN-CNN 模型，利用 textCNN 进一步提取评论文本的局部语法特征。在第二阶段的细粒度评价维度情感极性分类任务中，针对预训练模型字向量建模无法准确包含中文语义信息的缺陷，本文提出基于字词相似度加权的词向量构建方法，以 ERNIE 预训练模型生成的字向量与 Word2Vec 模型生成的词向量的相似度为权重对字向量加权求和构造词向量。此外，为进一步捕捉评论文本与第一阶段预测所得粗粒度评价维度之间的关联信息，本文采用注意力机制对评论文本与粗粒度评价维度的词向量作交互，构造基于注意力机制的评价维度词向量。

研究表明：①在粗粒度评价维度识别任务中，本文所提出的 LSAN-CNN 模型相较于原模型在粗粒度评价维度识别效果上有所提升，表明引入的 textCNN 模块能够增强模型的语义特征提取能力；②在细粒度评价维度情感极性分类任务中，本文所提出的基于字词相似度加权的词向量构建方法以及基于注意力机制的评价维度词向量均使得模型在方面级情感分析效果上有所提升，表明本文所采用的改进方案是有效的。

本文的创新点在于：①在 LSAN 模型基础上引入了 textCNN 模块，提出了 LSAN-CNN 模型；②基于中文领域词语包含的语义信息比单个汉字更为丰富这一前提，提出了基于字词相似度加权的词向量构建方法。同时本文仍然存在一些不足之处：①未针对数据均衡问题对模型稳健性的影响进行相关研究；②未将实

验结果与多任务模型作对比，以研究两阶段模式中存在的误差传导问题对模型分类效果的影响。

关键词：在线评论；方面级情感分析；多标签文本分类；预训练模型；ERNIE

目　　录

基于 LSAN-CNN 和 ERNIE 预训练模型的餐饮在线评论方面级情感分析

第一章　绪论

第一节　研究背景及意义

据中国互联网信息中心统计，截止 2020 年底我国的网民规模达到了 10.32 亿人，互联网普及率达到 73%。互联网的快速发展对人们的生活习惯也带来了各种各样的变化，网民抒发情感的渠道不再局限于线下，越来越多的用户选择在线发布自己对某个事物的评论来表达自己的意见或态度。而在餐饮行业，大量的餐饮消费在线评论文本也由此产生。对于消费者而言，发布的在线评论能够为其他消费者提供决策辅助，有些商家还会对发布优质评论的消费者提供优惠券等奖励，以此推动更多消费者积极发表评论；而对于商家而言，正面的在线评论能够提高自身的知名度和客流量，而负面的在线评论虽然会影响商家的排名，但也从侧面为商家提供了整改方向。当下，线上用户评论文本正呈现指数级增长的趋势，单纯依靠人工的方法对文本蕴含的情感信息进行挖掘存在耗时长且效率低下的问题。因此，互联网企业和相关研究人员开始关注如何利用算法自动化地提取评论文本中包含的有价值的信息，期间也催生了许多相关的技术方法。

情感分析（Sentiment Analysis, SA）是自然语言处理（Natural Language Processing, NLP）的重要研究方向之一，其主要任务是对文本数据中所包含的对某一客观事物的主观情感信息进行挖掘，也称为观点挖掘（Opinion Mining）。根据分析层次的不同，可将情感分析分为粗粒度情感分析和细粒度情感分析两大类，而方面级情感分析（Aspect-based Sentiment Analysis, ABSA）是细粒度情感分析的一个研究方向之一。根据文本中是否明确提到方面项，又可将 ABSA 任务分为基于方面项的情感分析（Aspect Term Sentiment Analysis, ATSA）和基于方面类别的情感分析（Aspect Category Sentiment Analysis, ACSA）[36]。餐饮消费在线评论一般包含有多个评价维度（即方面类别），且涉及的维度不一定显式地包含在文本中。例如，对评论“这家饭店太偏远了，但饭菜还不错，卫生条件也可以”来说，就包含了消费者对于该饭店的地理位置、食物口味和饭店环境三个维度的评价。其中对于位置的评价是消极的，而对于口味和环境的评价则是积极的。因此

对于餐饮消费评论而言，更适合对其进行 ACSA 任务。

随着互联网的发展和生活水平的提升，简单地将评论文本划分为单个极性的粗粒度情感分析对评论文本的信息挖掘不够深入，消费者和商家也都希望对产品或服务的各个方面有更全面的了解，因此粗粒度的情感分析不再能满足人们对文本数据挖掘的需求。方面级的情感分析可以对用户在线发布的评论作细粒度的情感分类，对海量的文本作结构化聚合，并在商家的线上 UI 界面进行展示。一方面，消费者可以据此了解到商家的产品在不同方面存在的优点与不足，进而作出消费决策；另一方面，商家可以通过评论的细粒度情感极性对产品各方面的优劣有更清楚的认知，进而有针对性地对产品进行整改，为消费者提供更优质的产品，进而增加经营利润。因此，将细粒度情感分析任务应用到在线评论中具有重要的研究意义。

第二节　国内外研究综述

一、多标签文本分类

文本分类是自然语言处理的经典任务之一，自上世纪 50 年代末开始就有针对该领域算法的研究[1]。传统的文本分类任务中，每个文本对应一个单独的类别标签，称为单标签文本分类（Single-Label Text Classification）。随着经济社会的发展，人们对待事物的观点和态度越来越多样化，使得文本内容日渐丰富，对文本分类领域的研究也从单标签文本分类扩展到了多标签文本分类（Multi-Label Text Classification）。在多标签文本分类任务中，单个文本可能对应一个或多个类别标签。因此，识别评论文本中所涉及的评价维度本质上是多标签文本分类任务。对多标签文本分类方法的研究大致可以分为两类：基于机器学习和基于深度学习的方法。

（一）基于机器学习的方法

从解决问题的策略上看，较为简单的方法是把多标签分类任务看作多个二分类任务的组合，而二分类任务已经有较多成熟的解决方案。Boutell 等（2004）[2]提出为每个标签构建一个单独的分类器来解决多标签分类的问题，该方法也称为二元关联法（Binary Relevance, BR）。BR 方法思路简单，但该方法潜在假定了标签之间是相互独立的，显然与实际应用场景存在一定差异。Tsoumakas 等（2007）[3]通过分析标签集的分布，将具有相同标签集的样本划分为同一类（Label

Powerset, LP），进而将多标签分类任务转化为多类别分类（Multi-Class Classification）任务。但该方法也存在明显的缺陷，即在标签数量较多时会导致数据稀疏性问题。针对 BR 方法忽略了标签之间相关性的问题，Read 等（2011）[4]提出分类器链（Classification Chain, CC）的概念，将多个二分类器串联成链式结构，每一个分类器的输入都依赖于上一个分类器的输出，依次调用完成多标签分类任务。

将多标签分类任务视为多个二分类任务的问题转化思想，是为了让数据适应现有的算法，反之也可以通过对现有算法进行改进来适应数据，这种思想称为算法自适应方法。Clare 等（2001）[5]等人通过对 C4.5 决策树算法进行改进，通过修改信息熵目标函数，提出适用于多标签分类任务的 ML-DT（Multi-Label Decision Tree）模型。Elisseeff 等（2001）[6]对传统的支持向量机（Support Vector Machine, SVM）模型进行改进，提出类似学习系统的排名支持向量机（Rank-SVM）模型用于多标签分类任务，并用核技巧（Kernel Trick）处理非线性情况下的分类任务。Zhang 等（2007）[7]提出 ML-KNN（Multi-Label K-Nearest Neighbor）模型，该模型借鉴传统 KNN 的思想，对于未观测样本先统计其最近邻样本的标签信息，而后基于最大后验（Maximum A Posterior, MAP）准则判断未观测样本的标签集。

（二）基于深度学习的方法

神经网络是一种在结构和功能上模仿生物神经结构的数学模型或算法，通过构造不同拓扑结构的神经网络能够对任意函数进行逼近。Zhang 等（2006）[8]首次将神经网络应用到多标签文本分类任务，提出一种适用于多标签文本分类的 BP-MLL（Backpropagation Multi-Label Learning）算法，该算法采用了新的损失函数用于捕捉多标签特征，并在基因功能分类和多标签文本分类任务中取得较优的效果。Nam 等（2014）[9]改进了 BP-MLL 算法，用交叉熵（Cross Entropy）损失函数替代了原有的排序损失函数，并在训练过程中采用 AdaGrad 优化器、Dropout 正则化方法和 ReLUs 激活函数，该方法在六个大规模多标签文本分类数据集上实现 SOTA（State-Of-The-Art）性能，同时也验证了交叉熵损失函数在多标签文本分类任务中的有效性。

然而，简单结构的神经网络在解决 NLP 任务时也存在着缺陷，一方面其无法保留文本完整的语义信息，另一方面其并未考虑到文本中单词的顺序。随着计算机硬件水平和算力的提升，基于深层次、复杂结构的神经网络方法即深度学习

方法开始应用于 NLP 领域，许多方法在文本分类任务中也取得了有效成果。Berger 等（2015）[10]分别使用 textCNN[11]和门控循环单元（Gate Recurrent Unit, GRU）对 Word2Vec[12]生成的词向量进行特征提取，最后根据人为设定的阈值判断标签类别。对于大规模多标签文本数据集而言，标签类别数过多造成的数据稀疏性是影响分类效果的一大问题，对此 Liu 等（2017）[13]提出 XML-CNN（Extreme Multi-Label CNN）模型。该模型在 textCNN 的基础上作了如下改进：一是使用动态最大池化（Dynamic Max-Pooling）代替原有最大池化，避免语义信息丢失的问题；二是使用交叉熵损失函数代替原有损失函数；三是在池化层和输出层之间加入全连接层，降低算法的复杂度。实验表明，该模型能够有效提升标签类别数较多的文本分类效果。CNN 独有的卷积核使其能够从局部到整体提取文本特征，但词语的顺序也是影响文本语义的关键要素，而 CNN 却未能对词序建模，在解决文本分类任务存在着局限性。循环神经网络（Recurrent Neural Network, RNN）相较于 CNN，在结构上更适用于处理序列数据。Chen 等（2017）[14]将 CNN 和 RNN 串联，将 CNN 提取到的文本特征再输入 RNN 层，相较于单个 CNN 层的模型在多标签文本分类任务上的效果得到提升。为了使得模型能够考虑到标签之间的依赖关系，又有学者将多标签文本分类任务转化为序列生成任务（Sequence Generation），序列生成模型也开始被应用于该领域。Nam 等（2017）[15]提出利用基于 RNN 的 Seq2Seq 模型进行多标签文本分类，在编码器端可以对词序建模，在解码器端又可以利用序列生成的思想对标签之间的关联性建模。在使用 Seq2Seq 架构进行多标签文本分类时，标签顺序对模型的性能会产生较大的影响。同样的标签集，模型会将不同顺序的集合判为负例。因此为减轻 Seq2Seq 模型对标签顺序的依赖性，Yang 等（2018）[16]在解码端增加了 Set-Decoder，提出了可用于多标签文本分类的 Seq2Set 模型，效果优于 Seq2Seq。为了更好地捕获不同单词之间以及标签与单词之间的关系，研究人员开始将注意力机制（Attention）加入到模型结构中。Lin 等（2018）[17][16]基于 Seq2Seq 架构，提出在编码器端采用多层扩展卷积（Multi-level Dilated Convolution, MDC）和长短期记忆网络（Long Short-Term Memory, LSTM）分别构造高层语义特征和词级别的语义特征，而后将解码器端的输出先后与编码器端 MDC 和 LSTM 的输出进行 Attention 操作（Hybrid Attention），兼顾了各个层级的特征信息，模型在路透社新闻数据集和中文博客数据集的多标签分类任务上取得 SOTA 结果。Xiao 等（2019）[18]提出

LSAN（Label Specific Attention Network）模型，将基于自注意力机制（Self Attention）生成的词向量和基于 Attention 生成的标签向量进行自适应融合（Adaptive Fusion），在中英文多标签文本数据集上都取得较好的分类效果。在 Attention 机制的基础上，Google 在 2017 年提出的 Transformer 网络架构[19]，对 NLP 领域产生了重要影响，也由此诞生了许多基于大规模语料的预训练模型。Yarullin 等（2020）[20]首次将 BERT 预训练模型[21]应用到多标签文本分类任务，提出序列生成 BERT 模型。

二、细粒度情感分析

作为 NLP 领域的一个细分任务，情感分析一直以来都受到广泛关注。早期的情感分析更多是从篇章级（Document Level）或句子级（Sentence Level）对文本蕴含的情感进行识别，其前提假设是整篇文章或单个句子只表达了一种积极或消极的情感[22]。而随着文本所包含信息的增多，人们对情感分析的要求也逐渐提高，情感分析开始从篇章级和句子级的粗粒度层面向细粒度层面发展。对细粒度情感分析任务的研究可以分为两类：一类是将粗粒度情感分析下的两类（积极/消极）或三类（积极/中性/消极）情感标签进行细化，将粗粒度的情感扩展为细粒度的情绪，如喜悦、愤怒、失望等；另一类是方面级（Aspect Level）的情感分析，即对文本中包含的评价实体或评价维度对应的情感极性进行识别，比如一段针对汽车的评论可能包含外观、内饰、动力、操控、空间等多个评价维度，每个维度所对应的情感类别又不尽相同。本文研究所涉及的细粒度情感分析任务属于第二类，即方面级的情感分析。针对细粒度情感分析的研究方法主要有三类：基于情感词典、基于机器学习和基于深度学习的方法。

（一）基于情感词典的方法

基于词典的情感分析方法需要先构建情感词典，进而根据词典中对应情感词的极性评分来判断文本的情感极性，因此情感词典的质量对情感分析的效果有很大影响。Hu 等（2004）[23]认为同义词具有相同的情感，而反义词的情感极性则相反，因此利用 WordNet 寻找同义词或反义词对人工构建的种子情感词典进行扩充，进而根据情感词与商品特征词的距离判断评论在特定商品维度下的情感极性。Moghaddam 等（2010）[24]基于 Epinions 大众消费点评网站构建情感词典，对于未知形容词则根据 WordNet 搜索，寻找 Epinions 网站已有词的相似形容词，

并对最邻近的词的评分进行加权作为未知形容词的估计评分。Cruz 等（2013）[25]在情感词抽取中融入领域特征信息，并使用 WordNet、PMI 算法和 SentiWordNet 进行情感分类。贾闻俊等（2016）[50]在抽取评论文本属性词基础上，采用 LDA 主题模型抽取情感词，进而基于情感词典判断特定属性下的情感极性。

（二）基于机器学习的方法

基于机器学习的情感分析方法需要利用特征工程提取文本数据相关特征，而后再用机器学习算法进行分类。Boiy 等（2009）[26]通过人工提取文本特征，进而采用 SVM、多项式朴素贝叶斯（Multinomial Naive Bayes, MNB）以及最大熵（Maximum Entropy）模型三种传统机器学习方法，在多语言文本数据集的细粒度情感分析任务上取得有效成果。Jiang 等（2011）[27]针对 Twitter 数据集构造了目标独立（Target-independent）和目标相关（Target-dependent）两类特征，其中目标独立的特征包括文本中的单词、标点符号、表情符号以及基于情感词典的特征，目标相关的特征则根据句法依存关系提取。最后基于 SVM 对 Twitter 文本在给定查询（query）向量下的情感极性进行分类，此处的 query 向量等同于方面级情感分析下的 aspect-term。Wagner 等（2014）[28]通过提取 N-gram 特征，加入基于情感词典以及方面项和情感词的距离计算得到的情感得分，采用 SVM 对餐饮和笔记本领域的评论数据集进行细粒度情感分析，取得了较优的分类效果。

（三）基于深度学习的方法

随着深度学习方法的发展，越来越多的研究着眼于将深度学习模型应用于情感分析领域，由此也衍生了许多针对情感分析任务的深度学习算法。对于细粒度情感分析任务，人们一般会根据目标词附近的上下文来判断该目标的情感类别。基于这一观点，Tang 等（2015）[29]提出目标依赖的 LSTM 模型（Target-Dependent LSTM, TD-LSTM），模型在目标词处将句子断开，进而用两个 LSTM 网络从正反两个方向作特征提取并进行拼接。但 TD-LSTM 模型并没有充分考虑目标词与评论文本的关系，因此 Tang 等（2015）[29]又提出目标连接的 LSTM 模型（Target-Connection LSTM, TC-LSTM）。该模型在 TD-LSTM 的基础上，在输入层将目标词向量与评论文本词向量进行拼接融合，随后由两个 LSTM 网络提取特征。实验显示，TD-LSTM 和 TC-LSTM 相较于 LSTM 模型，在方面级的细粒度情感分析任务中取得更高的准确率，而 TC-LSTM 对目标词与评论文本的融合也提升了分类的效果。为了更好地捕捉方面项和评论文本之间的关联性，Wang 等（2016）

[30]提出将 Attention 机制引入方面级情感分析中，提出 AT-LSTM（Attention-based LSTM）和 ATAE-LSTM（Attention-based LSTM with Aspect Embedding）模型。AT-LSTM 模型将 LSTM 提取的特征向量和方面向量进行拼接，然后采用 Attention 机制使模型重点关注对方面情感极性判断有较大影响的特征。而 ATAE-LSTM 在 AT-LSTM 的基础上，在输入层加入了方面词向量，进一步加强方面项与评论文本之间的关系。实验结果也显示，加入 Attention 机制的模型相较于 TD-LSTM、TC-LSTM 效果更好。MA 等（2017）[31]认为方面项与其上下文的关联性应该是相互的，因此提出 IAN（Interactive Attention Networks）模型，对方面项和上下文采用交互注意力机制提取特征并融合，在方面级情感分析任务中取得优于 ATAE-LSTM 的效果。随着 ELMo[32]、BERT[21]等预训练模型的出现，NLP 领域进入了预训练时代，研究人员开始将基于预训练模型的方案应用到情感分析任务中。由于 BERT 的输入可以是句子对，SUN 等（2019）[33]基于方面项构造了辅助句子，与评论文本形成句子对作为 BERT 的输入，将 ABSA 任务转化为类似问答或推断的句子对分类任务。作者采用了四种不同方式构造辅助句子，并对 BERT 进行微调，在两个细粒度情感分类数据集上实现 SOTA 效果。Jiang 等（2019）[34]将胶囊网络（Capsule Network, CapsNet）[35]与 BERT 结合，提出 CapsNet-BERT 模型，既利用了 CapsNet 捕捉方面项和评论文本之间的关系，又利用了 BERT 在大规模语料上学到的语义信息。

对现有文献进行梳理发现：

（1）对于多标签文本分类任务，早期的研究都是基于机器学习方法，解决问题的策略主要有两种：一是使数据适应算法，将多分类转换为二分类问题；二是使算法适应数据，即对现有算法进行改造。但机器学习方法的分类效果很大程度上取决于特征工程的质量，而传统的机器学习方法往往需要人工提取特征，因此特征工程的质量得不到有效保障。而基于深度学习的方法能够自动提取特征，省去了复杂的特征工程环节，也更适合当下数据规模日益增大的应用场景。因此，本文在 LSAN 模型结构的基础上进行改进，将其应用在评价维度的识别任务中。

（2）对于细粒度情感分析任务，早期的研究是基于情感词典计算得分的方法进行情感分类，该方法思路简单，但领域适用性较差。对不同领域的评论文本，需要单独构建针对该领域的情感词典才能取得比较好的分类效果。而基于机器学习的方法在情感分析任务中的准确率较基于情感词典的方法有所提升，但分类效

果同样也受人工提取特征质量的影响。随着深度学习算法的发展，越来越多的深度学习模型被应用到细粒度情感分类任务中，也取得了比传统方法更好的效果。但是，当前大多数深度学习模型都是在英文数据集上实现 SOTA 性能，专门针对中文数据集的细粒度情感分析模型相对较少。而在预训练模型的应用上，大部分研究都是利用预训练模型输出的字向量建模，而对于中文而言，词语所表达的语义信息要比单个字所表达的信息更有意义。因此，本文在评价维度识别任务的基础上，对百度 ERNIE 预训练模型生成的字向量加权构造词向量，并将评价维度信息融合到评论文本的语义特征中完成基于方面类别的细粒度情感分析任务。

第三节　研究思路及方法

一、数据来源

本研究采用的数据集是来自美团在 2021 年 4 月开源的 ASAP（Aspect Category Sentiment Analysis and Rating Prediction）数据集[36]，该数据集收录了美团旗下大众点评平台的合计 46730 条餐饮消费在线评论，经过人工判断标注细粒度情感极性，是目前基于方面类别的细粒度情感分析领域规模最大的中文数据集。该数据集在收录过程中遵从以下几个规则：①只收录字符数在 50~1000 之间的评论文本，不考虑较为极端的超短或超长文本；②不考虑非中文字符占比超 70%的评论文本；③过滤了广告等废文。这些规则保证了后续研究能够获得较高质量的数据。

二、数据预处理

由于数据集为网络文本，可能存在 HTML 标签、Unicode 乱码、表情等，标签或乱码并不具备语义信息，预处理阶段考虑删除。而表情虽然表达了情感信息，但在数据集中也是以 Unicode 编码存在，因此在构建模型输入时也不考虑表情符号。不同用户对输入法的使用习惯可能存在差异，导致评论文本中可能存在简体和繁体两种中文字体，而简繁体并不影响文字的语义信息，因此预处理时应将评论文本中的中文字体统一为简体，避免后续同一个汉字出现两种不同语义向量的情况。经过乱码处理、简繁转换之后，得到的是一份相对干净的文本数据集。而后便需要对文本进行分词，为词向量的训练构建输入数据。现有的分词工具都是基于其自带的字典，采用动态规划的方法进行分析，属于通用版本的分词，针对

特定领域的分词可能存在误差。考虑到本研究采用的数据集为餐饮消费领域，因此可以对分词工具原有词典进行扩充，增加搜狗饮食分类网络词库，提高餐饮领域评论文本分词的准确率。最后在预处理阶段，还需要去除文本中的标点符号以及停用词，保留评论文本主要信息。

三、基于 LSAN-CNN 的评价维度识别

对评论文本的细粒度情感分析，其目的是为了对评论中所涉及评价维度的情感极性进行分类，因此本研究将细粒度情感分析任务分为两个阶段完成。第一步是对评论文本中涉及的评价维度进行识别，第二步将评论文本与第一步中的评价维度识别结果进行融合，预测不同评价维度下的情感极性。

评价维度本质上属于多标签文本分类任务，该领域主流的研究方法都是基于深度学习算法，因此本研究在 LSAN 模型的基础上进行改进，对文本涉及的评价维度进行预测。LSAN 模型主要通过 Self-Attention 和 Label-Attention 机制分别生成两个维度相同的文本表征矩阵，进而采用自适应融合的方式对两个矩阵进行加权，得到最终分类器的输入。在文本特征提取阶段，LSAN 采用的是 Bi-LSTM，该方法可对文本的词序建模。在此基础上，本研究采用 textCNN 进一步提取文本的 n-gram 特征，得到文本的表征向量，与基于 Self-Attention 和 Label-Attention 得到的表征向量进行融合，最后输入分类器。

四、基于 ERNIE 的方面级情感极性分类

对于情感极性分类任务，本研究采用预训练模型 ERNIE。中文版本的预训练模型得到的是文本的字向量，以往的研究大部分也直接采用预训练模型产生的字向量作为文本的表征。但一个完整的中文词汇往往比单个汉字具备更丰富的语义信息，本研究将基于字和词的相似度，对字向量加权，构建词向量，进而与第一个任务中识别到的评论维度信息进行融合，作为最终情感极性分类器的输入。

模型对比方面，本研究将基于字词相似度加权的方法与直接采用字向量建模的方法进行比较，采用宏精确率、宏召回率和宏 F1 作为衡量模型性能的指标。

第四节　章节安排

本研究内容分为六个章节，各个章节安排如下：

第一章：绪论。首先对本研究的背景、目的及意义进行阐述，接着对细粒度

情感分析相关的国内外研究现状进行梳理，对现有研究中存在的问题进行分析，进而给出本研究采用的研究方法、创新点以及内容结构安排。

第二章：方面级情感分析相关理论方法。本章节主要介绍方面级情感分析的相关概念、模型理论和分析方法，包括当前常用的基于深度学习的词向量训练方法、自然语言处理领域常用的循环神经网络和卷积神经网络，并对注意力机制的计算方法以及基于 Transformer 架构的预训练模型进行介绍。

第三章：在线评论文本预处理。主要介绍模型所采用的数据集来源，由于数据集来源网络开源数据，因此数据的质量也需要有保障。本章将简要介绍官方披露的数据集采集过程中的一些细节问题，以证明该数据集具备较高的质量，能够为后续研究提供可靠保障。在此基础上，对数据集作必要的预处理，包括分词、去乱码、去停用词等步骤。

第四章：基于 LSAN-CNN 模型的评论维度识别。本章主要介绍细粒度情感分析的第一个任务——粗粒度评价维度识别，首先介绍用于多标签文本分类的 LSAN 模型的基本结构，对模型存在的不足进行分析，并给出本研究在 LSAN 模型基础上的优化思路，最后通过实验比较改进模型与原模型在评价识别任务中的效果验证该改进模型的有效性。

第五章：基于 ERNIE 的方面级情感极性分类。本章在评价维度识别任务的基础之上，进一步采用预训练的 ERNIE 模型并融合评价维度信息对评论的细粒度情感极性做分类。其中，将详细介绍本研究提出的基于字词相似度加权的词向量构造方法，并通过实验对比单纯使用预训练模型产生的字向量做分类的各项性能指标，以验证本研究提出方法的有效性。

第六章：总结与展望。本章将总结研究成果，在展望中分析本研究尚存在的不足之处以及未来的改进方向。

第五节　创新点与不足

一、创新点

本研究创新点主要为以下两个方面：

在评价维度识别任务中，本研究在梳理了多标签文本分类领域的研究现状后发现，当前主流的研究方法都是基于深度学习，在大规模中文多标签文本分类任务中，LSAN 模型取得了不错的效果。但是该模型在特征提取阶段只采用了 Bi-

LSTM 模型，无法提取到文本的局部特征，因此本研究拟增加 textCNN 模型，利用其卷积核的特性提取文本的 N-gram 特征，弥补 LSAN 模型在特征提取阶段的不足。

在细粒度情感极性分类任务中，梳理现有文献发现当前该领域主流的研究方法同样基于深度学习，而且随着预训练模型的发展，越来越多的研究采用“预训练+微调”的方式解决情感分析任务。基于中文语料的预训练模型针对字向量建模，因此大多数研究都直接采用模型生成的字向量作为下游模型结构的输入。但对于中文而言，词语的语义信息往往要比单个汉字更加丰富，也更有意义。基于此，本研究在运用中文预训练模型的同时提出了字词相似度加权构造词向量的方法，以弥补基于中文预训练模型的情感分析任务中单纯依靠字向量建模的不足。

二、研究不足

尽管本文研究所提出的改进方法能够提升模型在方面级情感分析任务上的效果，但研究过程仍然存在以下不足之处：①在数据预处理阶段，本文研究并未对数据均衡性作探究，标签不均衡可能对模型的识别效果造成一定的影响；②本文研究将方面级情感分析任务分为两阶段完成，该模式下会存在误差传导的问题，即第一阶段误差会影响第二阶段的效果，且本文并未将研究结果与多任务模型做对比。

第二章　方面级情感分析相关理论方法

对方面级情感分析领域的研究经历了从基于传统机器学习方法到基于深度学习方法的发展，而当前主流的研究方法大多都是基于深度学习中预训练模型的方法。对于本文研究所涉及的方面级情感分析任务，将采用深度学习方法，在文本向量化的基础上，使用深度学习网络对评论文本进行多标签文本分类，识别出所提及的粗粒度评价维度，进而将该评价维度信息与评论文本基于注意力机制作交互，并与经过字词相似度加权构建的评论文本词向量进行特征融合，完成细粒度评论维度上的情感极性分类。

本章将首先介绍研究过程中所涉及的相关概念，包括对多标签文本分类任务和方面级情感分析任务的内容进行阐述。其次将介绍 NLP 领域文本向量化表示的常用方法，包括主流的 Word2Vec 模型和预训练的语言模型 BERT。在此基础

上，本章还将介绍在提取文本关联信息方面应用广泛的注意力机制，详细阐述其原理和计算方法。最后，本章还将详细介绍深度学习方法中主要的特征提取器，包括循环神经网络、卷积神经网络和 Transformer 特征提取器。

第一节　相关概念

一、多标签文本分类

文本分类是 NLP 领域一个重要的研究方向，其主要目的是将文本划分为不同类别或分配不同标签。根据单个文本对应标签或类别数量的不同，文本分类又分为单标签文本分类和多标签文本分类。单个文本对应单个标签的任务称为单标签文本分类任务，而单个文本对应一个或多个标签的任务称为多标签文本分类任务。例如对一份新闻稿进行标签分类，因其内容可能由多则新闻构成，因此可能包含政治、体育、娱乐等多个标签，标签的数量至少为一个，属于多标签文本分类任务。

对于多标签文本分类任务，可将其表示为如下数学符号：记 $D=\{(x_i, y_i)|1\leqslant i\leqslant m\}$ 为训练集中的一个样本，其中 m 为样本总数，假设在该训练集上训练得到模型 $f: X\rightarrow Y$，其中 (X, Y) 为全部样本集，而 x_i 和 y_i 分别为样本集中的文本实例和对应的类别标签集合。则多标签文本分类任务可以表示为如图 2-1 所示。

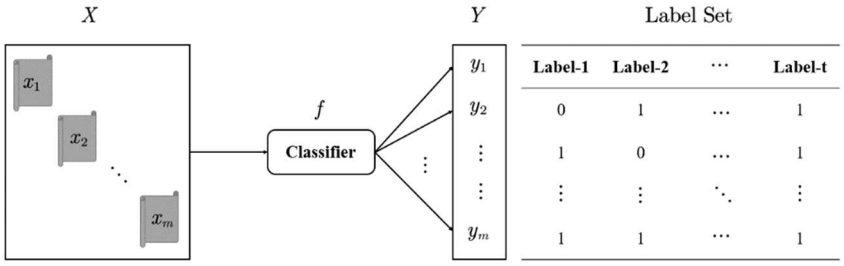

图 2-1　多标签文本分类框架

图中，训练集包含了 m 个样本，标签集 Y 中的每一个 y_i 都是一个 l 维的 0-1 向量，l 表示标签集中标签的数量，该表示方法相当于对标签集作了 One-Hot 编码。经由模型训练得到分类器 f，将给定文本 x_i 输入分类器 f 中得到对应标签集 y_i 即可完成多标签文本分类。若考虑数据处理过程，则完整的多标签文本分类任务步骤如图 2-2 所示。

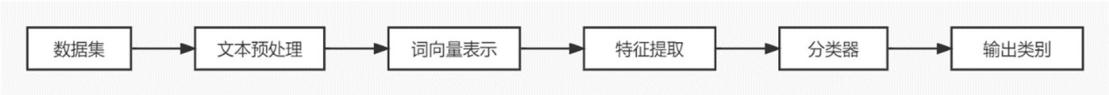

图 2-2　多标签文本分类流程

二、方面级情感分析

情感分析是 NLP 领域文本分类中的一个分支任务，其目的是对给定文本中某一客观事物的主观情感信息进行挖掘，因此也称为观点挖掘。根据分析层次的不同，情感分析又衍生出粗粒度情感分析和细粒度情感分析两个方向。在粗粒度情感分析任务中，单个文本仅对应一个情感类别。而在细粒度情感分析任务中，根据任务目标的不同又可划分为两类：一类是在粗粒度情感分析的基础上将粗粒度的情感扩充为细粒度的情绪，例如“喜”、“怒”、“哀”、“乐”、“惧”等；另一类是为单个文本中提及的不同方面或维度下的情感划分类别，也称为方面级情感分析。例如，对评论“这辆车动力很足，底盘调教也不错，但变速箱有点顿挫，体验感不太好”而言，该条针对汽车的评论就包含了对动力、底盘和变速箱三个方面的评价，其中对动力和底盘的评价是积极的，而对变速箱的评价是消极的。根据文本中是否显式地提及方面项，方面级情感分析又可进一步划分为基于方面项和基于方面类别的情感分析。由于餐饮消费在线评论文本不一定显式地包含方面项，因此本文研究所涉及的方面级情感分析任务属于后者。

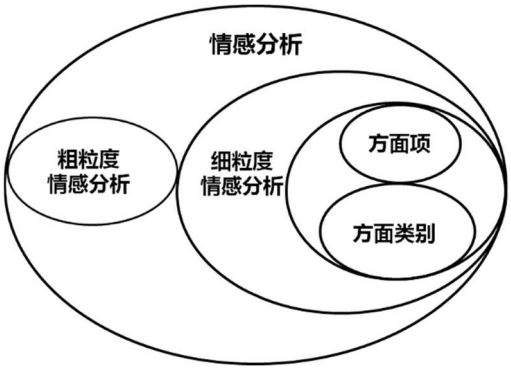

图 2-3　情感分析细分方向

假设评论文本所包含的方面类别或评价维度有m个，每个评价维度下又可划分为s个情感极性，则单个文本对应的情感类别实例可表示为如表 2-1 所示矩阵，矩阵的每一列为单个评价维度下由 One-Hot 编码表示的情感极性。该矩阵表示，评论文本在 Aspect-1 方面下的情感类别为 Sentiment-2，而在 Aspect-2 方面下的情感类别为 Sentiment-1，在 Aspect-m 方面下的情感极性为 Sentiment-s。

表 2-1　方面级情感分析标签集

	Aspect-1	Aspect-2	⋯	Aspect-m
Sentiment-1	0	1	⋯	0
Sentiment-2	1	0	⋯	0
⋮	⋮	⋮	⋱	⋮
Sentiment-s	0	0	⋯	1

第二节　文本向量化表示

互联网的发展使得信息的获取更加便捷和多样化，利用网络可以轻松获取到大量的文本数据。海量的文本数据蕴含着许多值得深入挖掘的信息，在线文本数据量的几何式增长也推动了对文本挖掘相关研究方法的发展。对于人类而言，对语言的理解能力是从小耳濡目染慢慢形成的，而对于机器而言亦是如此。要使得计算机能够理解自然语言，就需要在大量的文本语料上进行训练得到语言模型。而这一过程中至关重要的第一步，便是通过文本向量化表示技术将以文本形式存在的自然语言处理成计算机能够识别的数值形式。

按照生成方式的不同，文本向量化表示方法可分为离散化方法和分布式方法。离散化方法主要包含 One-Hot 编码、TF-IDF 编码、n-gram 表示等；而分布式方法主要是将单词映射为高维空间中的词向量，当前主流的分布式文本向量化表示方法都是基于深度神经网络实现的，包括 Word2Vec 模型、BERT 模型等，本文也主要介绍这两种文本向量化表示方法。

一、Word2Vec 模型

稀疏性和维度过高是传统离散化表示方法存在的缺陷，而分布式表示方法能够将词语映射为自定义维度的高维空间中的一个稠密向量，相较于离散化向量表示，分布式的词向量能够包含更多的语义信息，而词语之间的语义相似度可以通过词向量在高维空间中的距离来反映。常见的分布式词向量模型是 2013 年 Mikolov 等[12]提出的 Word2Vec 模型，其主要思想是利用语句中词与词之间存在上下文关联这一特点，构造神经网络完成词语预测词语的任务，属于一种无监督学习算法。根据预测方法的不同，又可将 Word2Vec 模型进一步分为 CBOW（Continuous Bag Of Words）模型和 Skip-Gram 模型。

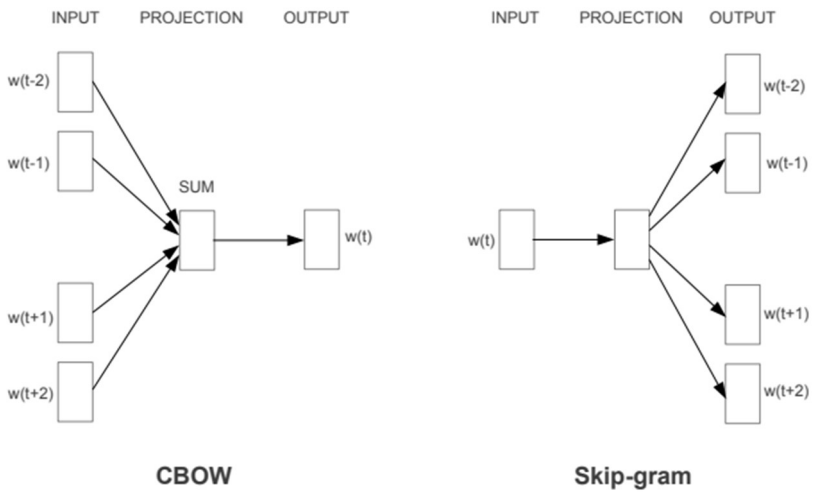

图 2-4 CBOW 模型和 Skip-Gram 模型框架[12]

（1）CBOW 模型

CBOW 模型的思想是利用某个中心词的上下文预测中心词出现的概率，由此得到的词向量自然包含上下文的语义信息。在模型中，需要指定窗口宽度，在指定宽度内位于中间位置的词即为中心词，其余词属于上下文，CBOW 模型任务就是基于极大似然估计方法，利用上下文输出词表中每个词出现在中心词位置的概率，将概率最大的词作为预测得到的中心词。CBOW 的具体模型架构如图 2-4 所示，图 2-4 展示了窗口大小为 2 的情况下模型的计算过程，即 w(t)作为待预测的中心词，其前后各两个词作为上下文输入模型。

记词表大小为V，窗口半径为s，则模型的输入由上下文的V维 One-Hot 编码向量构成，记为X，维度为$(2s, V)$。CBOW 模型的第一步是对输入的 One-Hot 编码矩阵映射到指定维度，该维度一般远小于词表的大小，此处记为$k(k \ll V)$，则线性变换矩阵W的维度为(V, k)。而后对上下文映射到k维空间的向量进行加总，得到中间向量H，维度为$(1, k)$。为了使得模型能够输出词表中每个词出现在中间位置的概率，需要将k维向量再次映射为V维向量，变换矩阵为W'，维度为(k, V)。不断迭代更新矩阵和W'，使得模型的交叉熵损失函数最小即为模型的最优解。其中，线性变换矩阵W即为在该语料上训练所得的k维词向量矩阵，第i行即为词表中第i个词对应的词向量。

（2）Skip-Gram 模型

Skip-Gram 模型的思想与 CBOW 模型相反，模型的输入是中心词，经过单层神经网络预测其上下文可能出现的单词，虽然预测方法不同，但模型得到的词向量同样包含了上下文的语义信息。Skip-Gram 模型架构如图 2-4 所示，图 2-4

展示了窗口半径为 2 的情况下模型的计算过程，其中 w(t)为中心词，作为模型输入预测其前后各两个单词。

与 CBOW 模型相同，训练模型的最终任务是得到线性变换矩阵。计算过程与 CBOW 模型计算过程类似，第一步先用(V, k)维度的变换矩阵W将输入矩阵映射为k维中间向量H，进而再利用(k, V)维度的变换矩阵W'得到输出V维的预测结果，即每一个单词出现在上下文的概率。对上下文单词的预测结果与真实上下文单词计算交叉熵损失，迭代更新变换矩阵W直至损失函数达到最小，得到词向量矩阵W。

在模型最后一步得到一个V维向量后，需要对其作 softmax 归一化计算才能作为概率。一般情况下词表大小V是一个比较大的值，而 softmax 归一化中又涉及指数运算，因此在V很大的情况下模型的每一次输入都需要很高的计算复杂度，模型训练效率不高。为此，Mikolov 等在 Skip-Gram 模型中提出了负采样（Negative Sampling）方法。负采样是一种提高数据集中负样本比例的方法，当负采样个数为n时，对于一个正样本就需要随机抽取n个负样本，达到$n+1$的样本量。应用到 Skip-Gram 模型中，在构造（中心词，上下文词）样本时，对于每一个正例样本，都从词表中随机抽取n个非上下文词与中心词构成负样本对，在最后 softmax 归一化时只在$n+1$维度上进行计算而不是词表维度V上，大大降低了模型计算的复杂度，提升了训练效率。

二、BERT 模型

词向量模型训练时所使用的语料规模越大，得到的词向量包含的语义信息就越丰富。因此为了得到更为准确的词向量，一般会将模型在大规模语料上进行训练。但庞大的数据规模对模型训练所使用的计算机硬件水平也提出了更高的要求，时间成本也更高，对于个人研究者而言实现难度大。而机构研究者和大型互联网公司得益于自身优越的硬件资源，能够轻松将模型在大规模语料数据集上进行分布式训练，得到包含通用语义知识的预训练模型。对于个人研究者而言，只需要使用预训练模型的参数进行初始化，用小规模数据集对模型参数进行微调，便可在下游任务中获得不错的效果，这就是 NLP 领域进入预训练时代后常用的“预训练+微调”范式。

2018 年，Devlin 等[21]提出了划时代的 BERT（Bidirectional Encoder

Representation from Transformers）模型，该模型主体架构由双向 Transformer 构成，经过在大规模语料上的预训练后，在下游 11 个 NLP 基准测试任务中达到了 SOTA 性能。传统的词向量模型如 Word2Vec、Glove 等均属于单向的语言模型，而 BERT 则采用了堆叠多层双向 Transformer 框架的方式作为特征提取器，构造双向语言模型，提高模型对文本语义信息的提取能力。

BERT 模型的输入主要由三部分构成，分别是词嵌入向量（Token Embeddings）、语句分块向量（Segment Embeddings）和位置编码向量（Position Embeddings）。词嵌入向量为输入的每个单词的 One-Hot 编码向量，而语句分块向量则标志单词对应所在的语句，位置编码向量用于表示单词所在语句中的位置。此外，输入语句中还包含“[CLS]”和“[SEP]”两个特殊标识符，“[CLS]”标志多个语句是否连贯，而“[SEP]”则作为不同语句之间的分隔符。

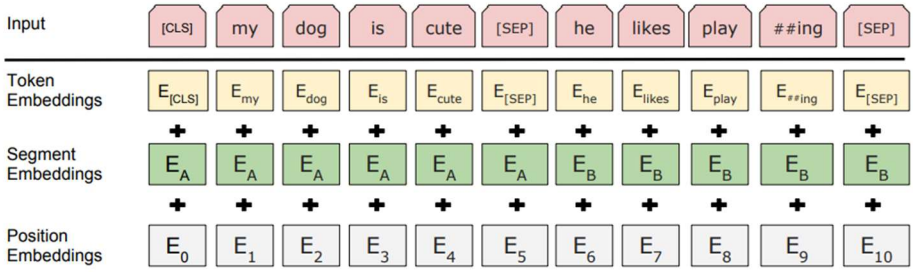

图 2-5 BERT 模型输入[21]

BERT 模型的预训练阶段主要有两个任务，分别是 MLM（Masked Language Model）任务和 NSP（Next Sentence Prediction）任务。MLM 任务类似于完形填空，先按照事先制定的 Mask 策略对输入文本进行随机 Mask 操作，进而再利用上下文预测经过 Mask 操作的单词。具体地，Mask 操作会将单词替换为“[MASK]”标识符，但该标识符仅在预训练阶段出现，下游任务中一般不会存在该标识符，会导致预训练阶段和微调阶段数据集存在偏差，因此不能简单地将单词替换为“[MASK]”标识符。因此 Devlin 等提出了如下 Mask 策略：①80%的概率替换为“[MASK]”标识符；②10%的概率替换为随机单词；③10%的概率保持不变。该策略也能够在一定程度上提升模型的鲁棒性。而对于 NSP 任务，其目的是判断多个语句是否连贯，也正因预训练阶段存在该任务，模型输入部分需要引入“[CLS]”标识符。加入 NSP 任务，能够提高模型在诸如智能问答、自然语言推理等下游任务上的适用性。

BERT 模型的简易架构如图 2-6 所示，特征提取层主要采用了双向

Transformer 框架，为了满足不同研究人员的需要，Devlin 等发布了 Base 版本和 Large 版本两种预训练模型。其中，Base 模型主要由 12 个 Transformer 编码器构成，而 Large 模型则包含 24 个，参数量达到了 340M，为 Base 版本参数量的三倍多，但模型的效果比 Base 版本也要更好。

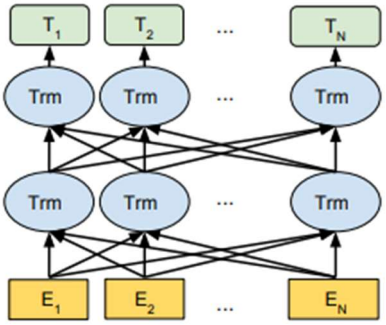

图 2-6 BERT 模型简易架构图[21]

第三节 注意力机制

注意力机制（Attention Mechanism）最早是应用在计算机视觉领域，其含义是人类在认知的过程中由于大脑资源受限而选择性地关注部分信息而非全部信息。例如当观察一张图片时，人们总是更多关注图片中色彩更为鲜艳或主体更加突出的部分。在自然语言的理解上也存在类似的现象，即当阅读一段文字时，对文本中的关键词会赋予更多的关注，因此研究人员也将注意力机制引入到 NLP 领域，在机器翻译、智能问答等领域得到广泛应用。本节主要介绍注意力机制的原理和计算方法，在此基础上再对多头注意力机制的原理作介绍。

一、注意力机制概述

Attention 机制本质上可以看作一种加权计算方法，假设存在查询向量Q、一组键向量K和一组值向量V，则 Attention 机制的操作过程便是将Q、K、V向量经过映射输出单个向量的过程。具体做法是对Q和每个K计算权重，利用该权重对每个V加权求和，得到最终的输出向量。

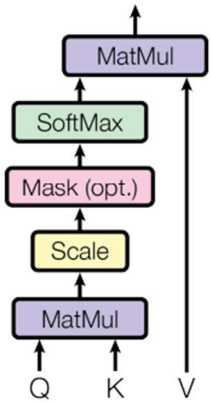

图 2-7　注意力机制计算流程[19]

具体计算过程如式 2.1 和 2.2 所示，对于查询向量Q，通过函数$softmax(Q,\ K_i)$得到基于查询向量与第i个键向量计算所得的权重，该权重可视为查询向量与键向量的相似度，也称为注意力分数（Attention Score）。利用该注意力分数对值向量V_i加权，得到基于 Attention 机制的输出向量。

$$a_i = softmax(a(Q,\ K_i)) \tag{2.1}$$

$$Attention(Q,\ K_i,\ V_i) = a_i V_i \tag{2.2}$$

式 2.1 中，注意力评分函数a的选取可以有多种，本文主要介绍加性注意力（Additive Attention）和缩放点积注意力（Scaled Dot-Product Attention）两种计算方法。在加性注意力中，评分函数a的定义如式 2.3，该函数将查询向量和键向量输入到一个多层感知机中，并使用不带偏置项的$tanh$函数作为激活函数。而在缩放点积注意力中，评分函数a如式 2.4 所示，其中d_k为键向量的维度，除以$\sqrt{d_k}$的操作便是对向量点积作缩放。

$$a(Q,\ K) = tanh(QW_Q + KW_K)W_V \tag{2.3}$$

$$a(Q,\ K) = \frac{QK^T}{\sqrt{d_k}} \tag{2.4}$$

二、多头注意力机制

多头注意力（Multi-head Attention）机制属于注意力机制的一类，但多头注意力并不直接对查询向量和键值对向量直接计算，而是先将Q、K和V作线性变换，如式所示，对线性变换后的查询矩阵和键值对矩阵作注意力分数的计算，将

这两个步骤同时进行h次，并将得到的结果作拼接，得到基于多头注意力机制的输出向量。

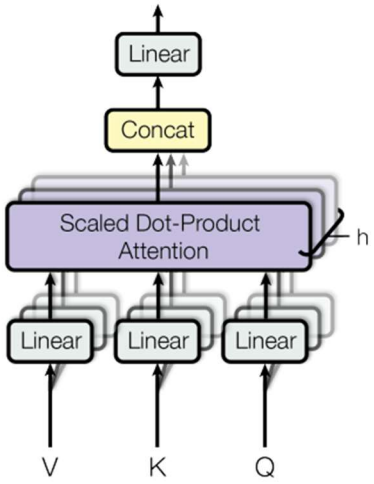

图 2-8　多头注意力机制计算流程[19]

多头注意力机制的具体计算过程如式 2.5 和 2.6：

$$\mathrm{MultiHead}(Q,\ K,\ V)=\mathrm{Concat}(head_1,\ head_2,\ \cdots,\ head_h) \tag{2.5}$$

$$head_i=\mathrm{Attention}(QW_i^Q,\ KW_i^K,\ VW_i^V) \tag{2.6}$$

其中，W_i^Q、W_i^K是维度为$(d_{model},\ d_k)$的线性变换矩阵，而W_i^V是维度为$(d_{model},\ d_v)$的变换矩阵，d_{model}为模型输入的向量维度。经过h次注意力计算后，将多个注意力计算结果进行拼接，并利用维度为$(h\times d_v,\ d_{model})$的变换矩阵将其映射为与输入向量相同维度的输出向量。为保证维度的一致性，需要满足以下条件：$d_k=d_v=d_{model}/h$。

由上述计算过程可以发现，多头注意力机制本质是进行多次独立的注意力计算，最终将多次计算结果集成到一个向量中，每个 head 中的注意力计算仅对输出序列中的单个子空间产生影响。因此，相比于单头注意力（Single-head Attention）机制，多头注意力机制能够在一定程度上防止过拟合。

第四节　特征提取器

传统的机器学习方法需要人工进行复杂的特征工程，才能在任务中取得不错的效果。而深度学习方法得益于丰富的网络拓扑结构，能够自动提取相关特征，相比传统机器学习方法效率更高。当前 NLP 领域所采用的深度学习研究方法中，循环神经网络、卷积神经网络和 Transformer 架构是常用的三大特征提取器。本

文研究在粗粒度评价维度识别任务中的模型所采用的特征提取器就包含了前两种，而在方面级情感分析任务中用到的预训练模型在特征提取层使用了Transformer，因此本节将对这三种特征提取器作详细介绍。

一、循环神经网络

循环神经网络（Recurrent Neural Network, RNN）[37]是深度学习领域中广泛使用的特征提取器之一，主要应用在序列数据建模领域。在自然语言中，上下文之间是存在关联性的，即当前位置的词与上一个词和下一个词存在相关性或依赖性。简单结构的神经网络并不能对词语之间的关联信息进行建模，在处理长文本的过程中也无法将前面的语义信息有效传递给后续的处理，即普通的神经网络不具备记忆功能。而 RNN 能够将上一时刻隐藏层的输出和当前时刻的输入同时传递到当前时刻的网络单元中，独有的网络结构使其天然具备时序建模的优势，因此 RNN 在 NLP 领域也得到了广泛应用。

根据输出层序列长度的不同，RNN 又可分为“N vs N”、“N vs M”、“N vs 1”和“1 vs N”四类。其中，输入和输出序列长度相同的网络属于“N vs N”类 RNN，适用于部分机器翻译应用场景；而对于“N vs M”和“1 vs N”类型的 RNN 则适用于智能问答、图片转文字等文本生成类场景；输出层仅为单个值的网络属于“N vs 1”类 RNN，其主要对最后一个时间步输出的隐藏层向量作线性变换后再输入到分类器中给得到单个计算结果，适用于文本分类场景。

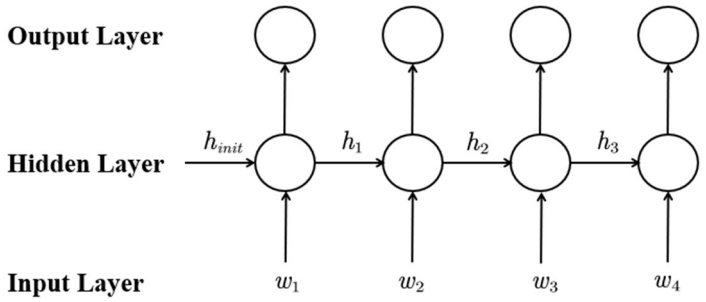

图 2-9 N vs N 循环神经网络框架

图 2-9 展示了“N vs N”类 RNN 的结构，此处假设文本包含四个单词，每一个时间步的输入x_t都有两部分构成，分别是上一时刻隐藏层的输出h_{t-1}和当前时刻的单词w_t。而对于初始时刻$t=1$，其隐藏层单元尚未有计算结果，因此需要指定一个初始化的隐藏层输出h_{init}代替。网络中每一个输入都对应一个输出，因此输入输出序列是等长的。

理论上 RNN 能够处理任何序列长度的时序数据，而实际应用过程中，RNN 在长序列建模时容易出现梯度消失和梯度爆炸问题。因为在反向传播的过程中，当前时刻t的梯度包含了所有后续梯度的乘积，在t比较小的情况下，若后续梯度很小则会发生梯度消失，而后续梯度很大又会造成梯度爆炸。长短期记忆网络（Long Short-Term Memory, LSTM）[38]的提出正是为了缓解 RNN 在处理长序列时存在的问题，其在 RNN 的基础上引入了细胞状态和门控机制，提出了输入门、遗忘门和更新门，实验也证明了加入门控机制的 LSTM 模型能够在长序列建模上比普通的 RNN 效果更好。

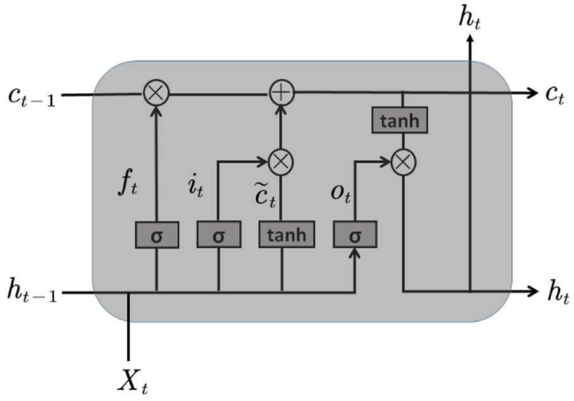

图 2-10　LSTM 模型计算过程

图 2-10 展示了 LSTM 网络中单个时间步的处理过程，其中c_t表示t时刻的细胞状态，$\tilde{c}_t$表示t时刻的候选细胞状态，而f_t、i_t和o_t分别表示遗忘门、输入门和输出门的值。遗忘门决定何时丢弃上一时刻细胞状态所传递的部分信息，这在时序较长时能够有效避免冗余信息的传递；输入门决定何时加入候选细胞状态的信息，即何时将新的信息往后传递；输出门负责将细胞状态转为隐藏层输出。具体计算方式如下：

$$f_t = \sigma(W_f \cdot [h_{t-1},\ x_t] + b_f) \tag{2.7}$$

$$i_t = \sigma(W_i \cdot [h_{t-1},\ x_t] + b_i) \tag{2.8}$$

$$\tilde{c}_t = tanh(W_c [h_{t-1},\ x_t] + b_c) \tag{2.9}$$

$$o_t = \sigma(W_o \cdot [h_{t-1},\ x_t] + b_o) \tag{2.10}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{2.11}$$

$$h_t = o_t \cdot tanh(c_t) \tag{2.12}$$

三种门控机制各司其职，使得 LSTM 模型既能够提取当前状态的信息，又能够对过去时刻的历史信息进行有效传递。LSTM 模型的这种特性表现在 NLP 领域便是其能够很好地对语句的时序信息进行建模，因为在自然语言中，语句的前后部分通常是存在一定关联的，而 LSTM 模型的记忆功能能够捕捉到上下文之间的关联信息，并将前文的有效信息传递给后文。

二、卷积神经网络

卷积神经网络（Convolution Neural Network, CNN）[39]也是深度学习领域常用的特征提取器之一，CNN 最早是应用在计算机视觉领域，其模仿人类观察图片的方式，使用卷积操作在图片上搜索低层次到高层次的特征，完成图片的自动化识别。而随后研究人员也将 CNN 引入到了 NLP 领域，并在文本分类、语义解析、情感分析等 NLP 任务上取得不错的效果。

CNN 主要由卷积层、池化层和全连接层构成，卷积层通过卷积核的滑动提取图片的局部特征，池化层负责对卷积操作提取到的特征进行降维处理，而全连接层负责将特征进行加权后输入分类器中。在文本分类领域，应用较广泛的 CNN 模型是 2014 年 Kim 提出的 textCNN 模型[11]，该模型的主要思想是利用多个不同维度的卷积核提取文本的局部信息，完成文本分类任务。textCNN 的模型架构如图 2-11 所示。

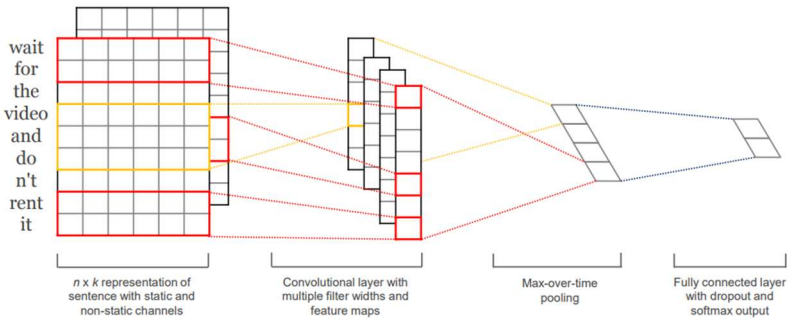

图 2-11 textCNN 模型框架[11]

假设X为输入文本，维度为(n, k)，其中n为句子长度，k为词向量维度。CNN 的卷积操作本质上是一种基于卷积核的映射过程，即利用参数矩阵将局部信息映射为单个值。在文本分类任务中，卷积操作是在句子长度维度上进行的，卷积核的宽度与词向量的维度保持一致，而高度则作为模型的超参数，因此卷积操作后得到的将是一个列向量。记卷积核的高度为f，滑动步长记为s，则卷积

操作后得到的向量维度计算公式为：

$$d = \frac{n-f}{s} + 1 \tag{2.13}$$

利用卷积核提取文本的局部信息得到特征向量后，将其输入到池化层中进行降维操作。根据计算方法的不同，一般可分为以下几种池化方法：①最大池化（Max Pooling），对每一种卷积核生成的特征向量取最大值；②平均池化（Average Pooling），对每个特征向量取平均值；③Top-K 池化（Top-K Pooling），对每个特征向量取前 k 个最大值；④Chunk-Max 池化（Chunk-Max Pooling），对特征向量分块后取每块中的最大值进行拼接作为池化结果。池化层能够将特征向量进行压缩，进一步提取关键的特征信息，简化后续的计算量。

三、Transformer

Transformer[19]是当前预训练模型中最常用的特征提取器之一，BERT、GPT 等大型预训练语言模型都是基于 Transformer 框架构建的，在多种 NLP 任务中也取得了很好的效果。Transformer 框架主要利用了多头注意力机制，其模型主要由堆叠的 N 个编码器和 N 个解码器两部分构成，模型架构如图 2-12 所示。

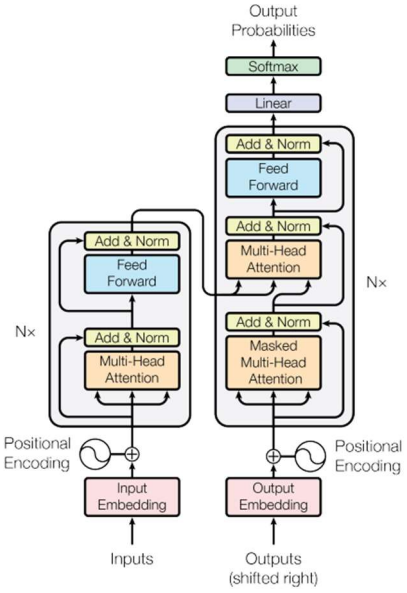

图 2-12 Transformer 模型框架[19]

如图 2-12 所示，左边部分为 Transformer 的编码器，右边部分为解码器。这种架构使得 Transformer 既可以适用于文本分类任务，也可以适用于文本生成任务。在分类任务中，只需要用到 Transformer 的编码器部分即可，在 BERT、ERNIE

等预训练模型中亦是如此，因此本文着重介绍编码器部分的计算过程。

Transformer 的输入部分由两部分构成，分别是词向量和位置编码向量。其中，词向量可以是由 Word2Vec、Glove 等词向量模型生成的预训练词向量，而位置编码向量则通过模型指定的位置编码函数将位置信息转换为嵌入向量。将词向量和位置编码向量对位相加，得到包含文本语义信息和位置信息的输入向量。而在编码器部分，Transformer 主要包含两个模块，分别是多头注意力模块和前馈神经网络模块。Transformer 中的多头注意力是基于自注意力（Self-Attention）机制实现的，普通的注意力机制需要确定Q、K和V三个向量，而自注意力机制中三个向量都来自输入向量，即$Q=K=V$。通过多头的自注意力机制使得编码器在对单个词语编码时能够关注到语句中其余单词的信息，充分提取句子中的上下文特征。而前馈神经网络层则负责对经过多头注意力机制的输出特征作线性变换，将特征映射到下游需要的维度上。

此外，由图 2-11 可以看到，编码器的每个子模块之后都紧随“Add & Norm”操作，其含义是残差连接和归一化操作。在 Transformer 框架中，其通过堆叠多个编码器构造深度神经网络提高模型的拟合效果，但过深的网络容易导致梯度消失等网络退化问题，而残差连接正是为了解决这一弊端而引入到模型中的。其次，归一化操作主要对每层的输入作归一化运算，防止因数值过大或过小而影响损失函数的收敛，保证模型训练的稳定性。

第五节　小结

本章主要对本文研究中方面级情感分析领域所涉及的相关概念和理论方法作了详细阐述。本文研究在粗粒度评价维度识别任务的基础上进一步完成方面级情感分析任务，因此本章首先介绍了多标签文本分类和方面级情感分析的概念和分类。在此基础上对情感分析任务中的第一步——文本向量化表示相关的研究方法作了介绍，主要包括传统的词向量模型 Word2Vec 和基于大规模语料的预训练模型 BERT。进一步又介绍了在文本关联信息提取方面应用广泛的注意力机制，对其原理和计算方法作了详细描述。在本章最后部分还介绍了情感分析任务中常用的三种特征提取器——RNN、CNN 和 Transformer，对每种特征提取器的原理和框架都作了详细论述。

第三章　在线评论文本预处理

模型的训练需要基于数值计算，而文字构成的语料并不能直接作为模型的输入，因此还需要经过一系列的预处理并将文本映射到高维向量空间后，才能作为模型的输入。本章节为数据预处理部分，为后续模型训练提供数据支持。本章节将首先介绍数据来源，简要描述数据采集过程中的相关细节，进而对该数据集进行清洗，最后基于 Word2Vec 模型对预处理后的语料进行词向量的预训练，并从词汇语义相似度方面考察了词向量模型的训练效果。

第一节　数据说明

一、ASAP 数据集概述

本文研究所采用的是美团在 2021 年 4 月开源的数据集 ASAP。该数据集收录了来自美团旗下“大众点评”平台上的 46730 条用户餐饮消费在线评论文本（其中包括训练集 36850 条，验证集 4940 条，测试集 4940 条），是当前方面级情感分析领域规模最大且质量相对较高的中文数据集。由于该数据集是专门针对基于方面类别的情感分析任务所提出的，数据集中预先定义了 18 个细粒度的评价维度，每个评价维度下又可划分出消极、中性、积极和未提及 4 种情感极性。为保证数据集的质量，在采集过程中遵循了以下原则：①剔除少于 50 个字符的短文本和多于 1000 个字符的冗长文本；②剔除非中文字符占比超过 70%的文本；③采用基于 BERT 的分类模型识别广告文本并予以剔除。而在标注过程中，美团采用了多轮标注和交叉比对的方法尽可能降低人工标注的主观性，提高标注的准确性。ASAP 数据集的高质量，保证了其能够为后续模型的训练和优化提供坚实的数据支撑。数据集的字段说明见表 3-1。

表 3-1　ASAP 数据标签说明

粗粒度评价维度	细粒度评价维度
位置(location)	交通是否便利(traffic convenience)
	距离商圈远近(distance from business district)
	是否容易寻找(easy to find)
服务(service)	排队等候时间(wait time)
	服务人员态度(waiter’s attitude)
	是否容易停车(parking convenience)
	点菜/上菜速度(serving speed)

价格(price)	价格水平(price level) 性价比(cost-effective) 折扣力度(discount)
环境(ambience)	装修情况(decoration) 嘈杂情况(noise) 就餐空间(space) 卫生情况(sanitary)
菜品(food)	分量(portion) 口感(taste) 外观(look) 推荐程度(recommendation)

二、ASAP 数据集可视化

对数据集中 18 个评价维度作统计分析得到图 3-1，可以看出，评论文本中涉及维度最多的是菜品的口味（Food#Taste），其次是服务的态度（Service#Hospitality）和价格水平（Price#Level）。

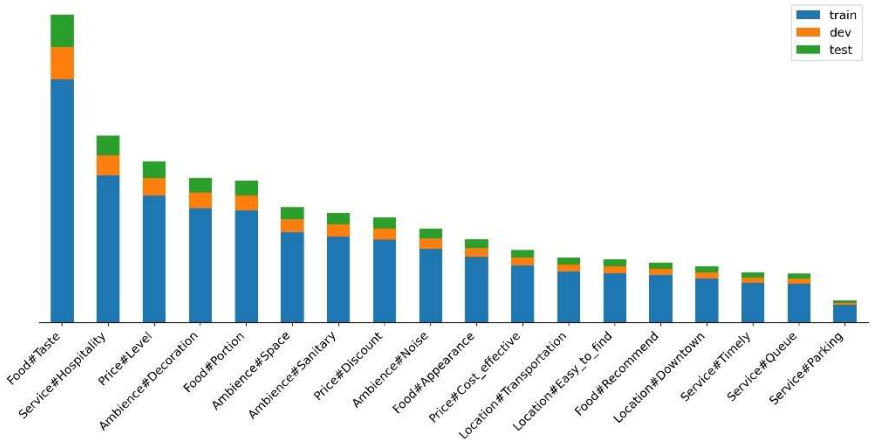

图 3-1　ASAP 数据集标签分布

对评论文本的长度作统计分析得到图 3-2，可以看出，60%左右的评论文本长度在 100~300 个字符之间，总体上看将近 90%的评论文本长度都在 500 个字符以内，该分析结果能够为后续模型训练中文本长度参数的设置提供一定的参考价值。

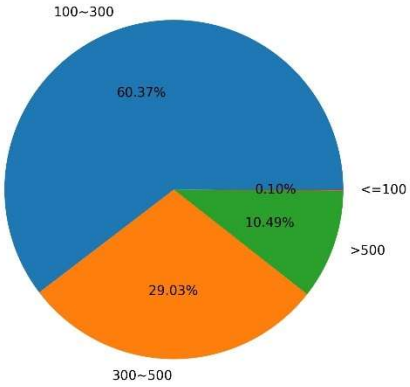

图 3-2　ASAP 数据集文本长度分布

第二节　数据预处理

数据集中的评论文本来源于线上平台的网络文本，因此需要对其进行一定步骤的预处理。本文主要对数据集做了如图 3-3 所示清洗：

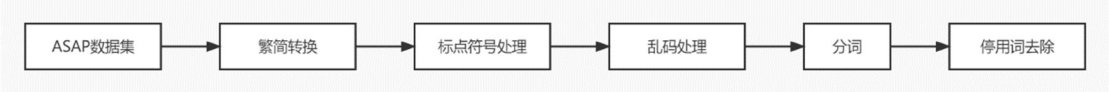

图 3-3　ASAP 数据集文本预处理流程

一、繁简转换

由于数据集来源于网络文本，不同用户有不同的输入法使用偏好，有的用户习惯使用繁体字输入法发表评论，因此数据集中存在着简体和繁体两种不同的中文字体。而从语义表示的角度上看，简体和繁体仅仅是书写方式的区别，其表达的语义信息是相同的。如果不对其作统一处理，则后续词向量的训练会出现冗余向量，即存在相同语义信息的不同向量表示，同时也会增大模型训练的时空复杂度，因此有必要对数据集中的简体字和繁体字进行统一表示。本文采用的处理方法是将繁体字转换为简体字，使用的是 Python 中的 zhconv 模块。

二、标点符号处理

出于语言表达的习惯，标点符号常用于作为语句划分标识。不同于正式的书面语言，网络文本中标点符号的使用相对较为随意，例如多个句号并排可能表示“无奈”的语气，亦或只是用户误输入了多个符号，实际语义信息并不明确。为了后续词向量训练的准确性，本文研究过程中统一对标点符号进行了剔除。

三、乱码处理

网络文本不可避免会存在 Unicode 乱码，而乱码是不具备任何语义信息的 Unicode 码，若不进行剔除，在词向量训练阶段模型会误将该乱码视为单个分词并赋予语义表征向量。因此，本文在数据预处理阶段对以下可能出现的乱码做了剔除，分别是\u0020（普通半角空格）、\u3000（普通全角空格）、\u00A0（html 实体不间断空格）、\xa0（十六进制 html 空格）、\u2002（html 实体半角空格）、\u2003（实体全角空格）等。

四、分词

经过上述步骤的清洗后，还需要对评论文本进行分词，以便后续词向量模型的训练。分词工具方面，本文选取的是 Python 的 jieba 中文分词库，其分词原理大致如下：首先基于自带的默认词典构建前缀词典，进而根据输入文本扫描前缀词典的结果构建有向无环图（DAG），最后基于动态规划在 DAG 中寻找概率最大的路径得到分词结果。jieba 的默认词典属于通用领域词典，而本文研究所采用的数据集来源于餐饮消费领域，为了提高分词的准确性，本文在分词阶段采用了搜狗网络词库中“饮食”类目下的多个词库对 jieba 库的默认词典进行了扩充，扩充前后分词效果比较见表 3-2。

表 3-2　词典扩充前后分词结果对比

原始文本	他家的冰糖乳鸽燕窝羹非常值得推荐，松鼠鱼也很新鲜
词典扩充前分词结果	他家 的 冰糖 乳鸽 燕窝 羹 非常 值得 推荐 ， 松鼠 鱼 也很 新鲜
词典扩充后分词结果	他家 的 冰糖乳鸽燕窝羹 非常 值得 推荐 ， 松鼠鱼 也很 新鲜

五、停用词去除

由于中文表达的习惯，评论文本中含有大量的“的”、“地”、“得”等虚词，也称为停用词（Stop Words）。在词向量训练时，无需关注这一部分词语，因此可以在预处理阶段对停用词进行剔除。本文综合了目前开源的多个停用词表，包括

中文停用词表、百度停用词表、哈工大停用词表和四川大学机器智能实验室停用词库等，整合成为新的停用词表，尽可能完整地去除数据集中的停用词。

第三节　词向量训练

由于 NLP 领域中的语料大多都属于未标注数据，人工标注的语料往往不易获得，因此当前主流的词向量训练方法便是采用无监督学习的词向量模型对未标注语料作预训练，而 Word2Vec 模型在词向量训练领域已经得到广泛应用，模型较为成熟，因此本文选取该模型作为词向量模型进行训练。

经过上述步骤的预处理后，得到的是分词后的 ASAP 数据集，数据集的基本情况如表所示。由于本文研究的领域为餐饮消费领域，仅仅采用了 ASAP 数据集作为词向量训练语料，后续再根据具体的任务对词向量作微调，该做法能够使得模型学到的语义向量领域适用性更强。

表 3-3　ASAP 数据规模概览

评论数（条）	预料大小（GB）	总词汇量（万）	词表大小（万）
36850	0.11	390.5149	12.7415

一、模型参数

工具方面，Python 的第三方开源库 gensim 包含了 Word2Vec 子模块，该模块内置了 CBOW 模型和 Skip-Gram 模型，本文选取的是 Word2Vec 模型的 CBOW 模式对词向量作预训练，模型的输入是经过分词的评论文本，CBOW 模式下 gensim 库会自动划分出窗口内的中心词和上下文，构造出训练集和对应标签，调用时只需要设置好相应的参数即可。

模型训练过程中，相关参数设置如下：①参数 size 用于设置模型输出词向量的维度，若维度过低，则词向量无法完整地包含词语的语义信息，而维度过高又会增加后续模型训练的时空复杂度。本文在该阶段获得的词向量将作为后续模型输入的初始化词向量，因此词向量的维度应当与后续模型设置相对应，此处 size 设置为 768；②参数 window 用于设置当前词与目标预测词之间的最大距离，若采用 CBOW 模型，则目标预测词即为滑动窗口内的中心词，而 window 相当于滑动窗口的半径。此处保持仍模型默认的设置，window 为 5；③参数 min_count 用于设置单词最少出现次数，出现频率低于 min_count 的单词将会被丢弃，该操

作会减小最终训练所得词表大小。由于本文在评论文本预处理阶段已经将乱码和停用词进行了清洗，在词向量预训练阶段不再对词表的筛选作过多的限制，min_count 设置为 1；④参数 iter 设置模型迭代次数，经过多次实验比较词向量训练效果，模型迭代 10 次时效果相对较好；⑤参数 alpha 和 min_alpha 是对学习率衰减策略的设置，alpha 设置为 0.025，min_alpha 设置为 0.0001，则训练过程中学习率会从 0.025 线性下降为 0.0001；⑥参数 negative 设置负采样的个数，采用默认设置 5。

表 3-4 Word2Vec 模型参数设置

参数	参数值
size	768
window	5
min_count	1
iter	10
alpha	0.025
min_alpha	0.0001
negative	5

二、训练效果

将分词后的训练集语料作为模型的输入，经过 10 次迭代训练后，得到 127415 个词向量。此外，还需要增加两个特殊的词向量，分别代表“[PAD]”和“[UNK]”。其中，“[PAD]”主要作为后续模型训练时将评论文本补齐到统一固定长度所采用的标识符，而“[UNK]”主要是对样本外数据集进行预测时用于表示未登录词的标识符。在词向量的预训练阶段，本文对这两个特殊标识符赋予了分别赋予了 0 向量，后续可以根据具体任务参与微调。

```
word embedding of "薯条" :
array([ 6.13049626e-01, -5.44995308e-01,  1.16469622e+00,  4.18366939e-01,
        9.54990149e-01,  2.66225457e-01,  1.39576465e-01, -2.14281499e-01,
       -2.32250378e-01,  3.53757650e-01,  1.43504775e+00, -2.50038922e-01,
       -1.80111796e-01, -6.72532991e-02,  1.65632516e-01, -1.36010301e+00,
        1.15622014e-01,  1.12847114e+00,  1.17524289e-01, -6.65623903e-01,
       -7.65461743e-01,  2.96926856e-01,  3.45769435e-01,  1.19358324e-01,
       -1.30304372e+00, -4.89617795e-01,  6.42624319e-01,  1.08907327e-01,
       -6.53605938e-01,  4.13434327e-01, -1.12537348e+00,  2.92011976e-01,
       -3.97083193e-01, -5.06431982e-02,  3.62939656e-01, -1.94632903e-01,
        2.86620587e-01, -1.88381866e-01, -6.97584927e-01, -1.15551138e+00,
       -3.78080100e-01,  3.10999423e-01, -1.41341054e+00,  5.25649667e-01,
       -1.55006754e+00, -1.35963678e+00,  1.27998829e-01, -3.72039497e-01,
        7.92088866e-01, -7.17891574e-01,  5.06971657e-01,  4.43602681e-01,
        ... ...], dtype=float32)
```

图 3-4 词向量训练结果实例

在传统语法规则中，词汇语义相近或相同的词语称为近义词，而在词汇经过

词向量模型映射为高维空间向量后，词汇之间语义的近似程度理论上表现为词向量之间的距离，语义信息越接近则两个词向量应当距离越近。Word2Vec 模型提供了不同词汇之间基于余弦相似度的计算，同时可以查询与某个词汇相似度最高的前 k 个词。此处以词语“薯条”为例，与“薯条”该词的词向量最为接近的 Top-10 个词向量如表 3-5 所示。

表 3-5　与“薯条”最相似的前 10 个词语

词语	相似度
“鸡米花”	0.852243959903717
“薯角”	0.7987374067306519
“薯格”	0.7732587456703186
“粗薯”	0.7457643747329712
“番茄酱”	0.7324262857437134
“炸薯条”	0.7253045439720154
“汉堡”	0.6886316537857056
“炸鸡块”	0.6879271268844604
“薯饼”	0.6708752512931824
“炸鸡”	0.6634553074836731

可以看出，与“薯条”最相似的前十个词向量均为快餐店的食品，而“薯条”本身也是出现在快餐店居多，该组词语所代表的食品与“薯条”是快餐店经常出现的搭配，因而与“薯条”出现在同一条评论中的可能性也较大，而“番茄酱”作为“薯条”的蘸料也排在了前 5 的位置，从直观上判断该组词汇与目标词汇之间确实存在较强的语义关联，这也表明基于 ASAP 数据集的语料学到的信息能够较好地反映词汇之间的语义相似性，词向量模型的训练效果较好。

CBOW 模式下的 Word2Vec 模型基于未标注的语料进行无监督学习，利用滑动窗口内的上下文对中心词进行预测，其前提假设是上下文与中心词之间存在关联性，这也符合日常语言表达的语法习惯，因此模型能够充分学习到语料中包含的语义信息，为后续其他任务模型的训练提供更好的初始化向量。

第四章　基于 LSAN-CNN 模型的评价维度识别

在 ASAP 数据集中，本文从 18 个细粒度评价维度中抽取出 5 个粗粒度评价维度，分别是价格（Price）、位置（Location）、服务（Service）、环境（Ambience）和菜品（Food）。本文将在这 5 个粗粒度评价维度识别任务的基础之上，将所涉及粗粒度评价维度的语义信息融合到评论文本中完成情感极性分类，因此评价维

度识别任务是整个方面级情感分析任务的第一步。单个评论文本所提及的评价维度可能有一个或者多个，因此该任务本质上属于多标签文本分类任务，而当前主流的解决方案都是基于深度学习，因此本文也选取了多标签文本分类领域效果相对较好的 LSAN 模型作为基础模型进行改进，完成对评论文本评价维度的识别。

本章将在得到预训练词向量的基础上，进一步对评论文本中所提及的评价维度进行识别。内容方面，本章将先对 LSAN 模型作更加详细的介绍，并分析现有模型存在的缺陷，进而提出改进的方法，构建 LSAN-CNN 模型，旨在更加全面地提取评论文本的语法特征，提高评价维度识别的效果。最后通过在 ASAP 数据集上进行实验，比较模型改进前后对评价维度的识别效果，验证本文所提出的模型改进的有效性。

第一节　LSAN 模型分析

为单个文本分配一个或多个标签的任务称为多标签文本分类，对该领域的研究经历了从传统机器学习方法到深度学习方法的发展。在多标签文本分类领域中，标签的数量一般不少于三个，部分场景下标签数量可达几十甚至上百个，要准确实现标签的识别存在一定的难度。在 Attention 机制引入该领域研究后，Xiao 等在 2019 年提出了 LSAN（Lable Specific Attention Network）模型[18]，旨在通过捕捉标签与评论文本的关联性提高分类效果。模型整体的框架如图 4-1。

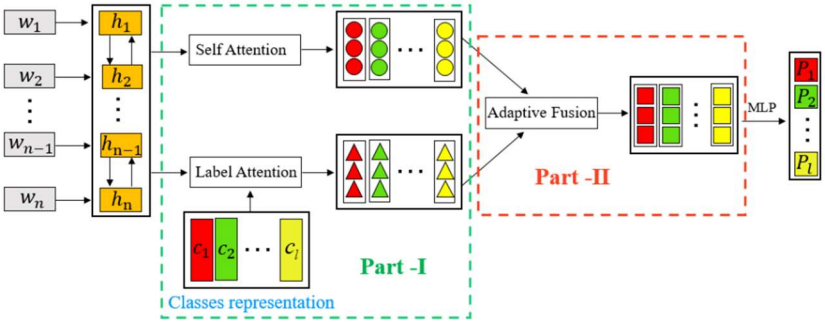

图 4-1　LSAN 模型框架[18]

一、LSAN 模型框架

LSAN 模型分为两个部分，第一部分主要从评论文本以及标签集合中挖掘相关的语义特征，第二部分对第一部分得到的特征进行自适应融合。模型输入方面，LSAN 模型的输入也由两部分构成，首先是文本的词向量矩阵，其次是标签词向量矩阵。在 LSAN 模型中，标签被认为与文本中的单词一样具有自身的语义信

息，因此也应当为每个标签赋予一个词向量。记第i条文本的词向量矩阵为$x^{(i)}$，其中第j个词的词向量为$w_j^{(i)}$，则$x^{(i)}=\{w_1^{(i)},\ w_2^{(i)},\ \cdots,\ w_n^{(i)}\}$，维度为$(k,\ n)$，其中$k$为词向量维度。记标签词向量为$C=\{c_1,\ c_2,\ \cdots,\ c_l\}$，即共有$l$个标签，词向量维度与文本词向量维度保持一致，因此$C$的维度为$(k,\ l)$。

在特征提取层，LSAN 模型采用了双向 LSTM（Bi-LSTM）网络提取语义特征，得到隐藏层特征向量。记从左至右第p个时间步的隐藏层向量为$\overrightarrow{h_p}$，从右至左第p个时间步的隐藏层向量为$\overleftarrow{h_p}$，则有

$$H^{(i)}=\mathrm{Concat}\left(\overrightarrow{H^{(i)}},\ \overleftarrow{H^{(i)}}\right) \tag{4.1}$$

$$\overrightarrow{H^{(i)}}=\left(\overrightarrow{h_1^{(i)}},\ \overrightarrow{h_2^{(i)}},\ \cdots,\ \overrightarrow{h_n^{(i)}}\right) \tag{4.2}$$

$$\overleftarrow{H^{(i)}}=\left(\overleftarrow{h_1^{(i)}},\ \overleftarrow{h_2^{(i)}},\ \cdots,\ \overleftarrow{h_n^{(i)}}\right) \tag{4.3}$$

其中，$H^{(i)}$表示第i条评论文本对应的隐藏层特征向量构成的矩阵，维度为$(2k,\ n)$。

为了更加充分地捕捉单条输入文本中单词之间的关联信息，对经过Bi-LSTM网络提取的隐藏层矩阵$H^{(i)}$作自注意力（Self-Attention）操作。首先需要计算注意力得分，其中，W_1和W_2分别是维度$(d_a,\ 2k)$和$(l,\ d_a)$的参数，d_a为模型中的超参数，可以自行设定。$A_s^{(i)}$为第i条文本所对应的注意力得分矩阵，维度为$(l,\ n)$。将该得分矩阵$A_s^{(i)}$与隐藏层矩阵$H^{(i)}$相乘，得到基于 Self-Attention 机制加权的特征矩阵$M_s^{(i)}$，维度为$(l,\ 2k)$。

$$A_s^{(i)}=softmax\left(W_2\cdot tanh\left(W_1H^{(i)}\right)\right) \tag{4.4}$$

$$M_s^{(i)}=A_s^{(i)}\left[H^{(i)}\right]^T \tag{4.5}$$

除了对特征矩阵作 Self-Attention 操作外，更为重要的是捕捉到标签与文本之间的语义关联信息，这也是模型名称中“Label Specific”的由来。在 LSAN 模型中，通过计算标签词向量与 Bi-LSTM 隐藏层向量之间的相关性作为权重，对 Bi-LSTM 提取的特征向量进行加权，该步骤在模型中称为 Label-Attention，最终得到基于 Label-Attention 机制加权的特征矩阵$M_l^{(i)}$，维度为$(l,\ 2k)$。

$$\overrightarrow{A_l^{(i)}}=C^T\,\overrightarrow{H^{(i)}}\qquad\overleftarrow{A_l^{(i)}}=C^T\,\overleftarrow{H^{(i)}} \tag{4.6}$$

$$\overrightarrow{M_l^{(i)}}=\overrightarrow{A_l^{(i)}}\left[\overrightarrow{H^{(i)}}\right]^T\qquad\overleftarrow{M_l^{(i)}}=\overleftarrow{A_l^{(i)}}\left[\overleftarrow{H^{(i)}}\right]^T \tag{4.7}$$

$$M_l^{(i)} = \text{Concat}\left(\overrightarrow{M_l^{(i)}}, \overleftarrow{M_l^{(i)}}\right) \tag{4.8}$$

此处，两个加权特征矩阵的维度是一致的。在第二部分，LSAN 模型对特征矩阵$M_s^{(i)}$和$M_l^{(i)}$进行融合，采用的是自适应融合（Adaptive Fusion）策略。自适应融合本质上也是加权融合，不同的是权重α和β作为可变参数，可以随着模型训练迭代更新。此外，α和β也反映了基于 Self-Attention 和 Label-Attention 得到的特征矩阵的重要性，理论上两个权重之和应为 1，模型也作了相应的限制，其中W_3和W_4是维度为$(2k,\ l)$的参数向量。

$$a^{(i)} = \sigma(M_s^{(i)} W_3) \quad \beta^{(i)} = \sigma(M_l^{(i)} W_4) \tag{4.9}$$

$$\alpha^{(i)} + \beta^{(i)} = 1 \tag{4.10}$$

$$M^{(i)} = \alpha^{(i)} M_s^{(i)} + \beta^{(i)} M_l^{(i)} \tag{4.11}$$

融合的特征矩阵经过两个全连接层后，输入到分类器中。对于多标签文本分类任务，单条输入文本情况下最后输出的是维度与标签数量相同的向量，向量的每一个元素代表文本属于对应位置标签的概率，本质上是多个二分类器的组合，因此在分类器层只需要将 sigmoid 函数应用到经过全连接层降维后的特征向量的每一个元素上即可。其中，W_5是维度为$(b,\ 2k)$的参数，b为模型的超参数，而W_6是b维的向量，最终得到的预测值$\hat{y}_i$是一个l维向量。

$$\hat{y}_i = \sigma\left(W_6 \cdot f\left(W_5 [M^{(i)}]^T\right)\right) \tag{4.12}$$

二、LSAN 模型损失函数

在二分类任务中，损失函数为交叉熵损失函数，而多标签文本分类又可以看作多个二分类任务的组合，同样可以使用交叉熵损失函数作为优化的目标函数。其中，y_{ij}和$\hat{y}_{ij}$分别为第i条评论文本的第j个标签的真实值和预测值，$y_{ij} \in \{0,\ 1\}$，$\hat{y}_{ij} \in [0,\ 1]$。

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{j=1}^{l}\left[y_{ij} log \hat{y}_{ij} + (1 - y_{ij}) log\left(1 - \hat{y}_{ij}\right)\right] \tag{4.13}$$

第二节　模型改进

一、LSAN-CNN 模型框架

LSAN 模型在结构上具有两个创新之处：其一是提出了 Label-Attention 机制，通过引入标签词向量充分利用了标签集所包含的语义信息，进而对标签集与文本之间的关联信息进行提取构造特征矩阵；其二是引入了 Adaptive-Fusion 机制，使得对基于 Self-Attention 和 Label-Attention 得到的两个特征矩阵进行加权时可以自动调整权重。得益于这两种机制的提出，LSAN 模型在 RCV1、AAPD、EUR-Lex 和 KanShan-Cup 四个中英文大规模多标签文本分类数据集上取得了不错的分类效果。

在特征提取层，LSAN 模型采用了 Bi-LSTM 网络。单向的 LSTM 网络可以按照语序从句首到句末提取语义信息，而双向的 LSTM 网络得到的特征则同时包含了前向和后向的语义信息，相比于单项的 LSTM 在特征提取上更为全面。但是，对于文本的局部语法特征（n-gram），LSTM 网络并不能很好地捕捉。因此，本文提出 LSAN-CNN 模型，该模型在 LSAN 模型的特征提取阶段加入了 textCNN 作为局部语法特征提取器，旨在弥补 LSAN 模型在特征提取上存在的缺陷，提高模型整体的分类效果。

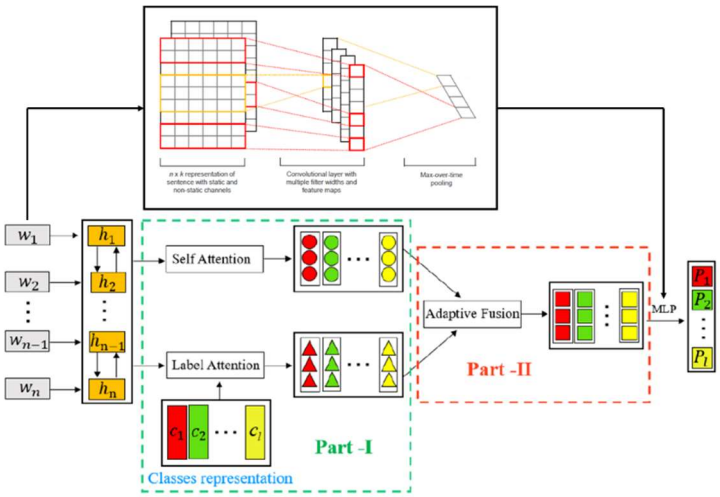

图 4-2　LSAN-CNN 模型框架

二、LSAN-CNN 模型原理

模型的输入部分仍然是预训练的文本词向量和标签集词向量，不同的是，除了经过 Bi-LSTM 层提取特征外，还需要同时经过 textCNN 层进行特征提取。记

第i条文本的词向量矩阵为$x^{(i)}=\{w_1^{(i)},\ w_2^{(i)},\ \cdots,\ w_n^{(i)}\}$，其中$w_j^{(i)}$为第$j$个词的词向量，则$x^{(i)}$的维度为$(k,\ n)$。在 textCNN 中，卷积核的宽度与词向量的维度保持一致，而高度则自行确定，此处卷积核高度设为$(f_1,\ f_2,\ \cdots,\ f_s)$，而每种尺寸的卷积核数量设为$(N_1,\ N_2,\ \cdots,\ N_s)$，卷积步长设为 1。记第$i$条文本对应的第$m$种卷积核尺寸下的卷积结果为$C_m^{(i)}$，$C_{mt}^{(i)}$为第$m$种卷积核第$t$次卷积计算的结果。经过多次卷积操作，得到多种不同维度的列向量，其中$C_{mt}^{(i)}$的维度为$(n-f_m,\ 1)$。卷积操作之后，需要对每种卷积核得到的结果进行降维，本文采用的是最大池化（Max-Pooling）方法将每个每次卷积计算得到的列向量$C_{mt}^{(i)}$压缩为单个数值，最后将 Max-Pooling 得到的多个数值拼接为特征向量，维度为$(1,\ N_1+N_2+\cdots N_s)$。

$$C_m^{(i)}=\text{Convolution}\left(x^{(i)}\right) \tag{4.14}$$

$$C^{(i)}=\text{MaxPooling}\left(\underbrace{C_{11}^{(i)},\ \cdots,\ C_{1N_1}^{(i)}}_{N_1},\underbrace{C_{21}^{(i)},\ \cdots,\ C_{2N_2}^{(i)}}_{N_2},\ \cdots\ \cdots,\underbrace{C_{s1}^{(i)},\ \cdots,\ C_{sN_s}^{(i)}}_{N_s}\right) \tag{4.15}$$

为使得 LSAN 模型中的特征向量和加入 textCNN 后得到的特征向量能够进行融合，在最后的全连接层也需要对 LSAN 模型进行改动，原 LSAN 模型将基于 Self-Attention 和 Label-Attention 得到的融合特征矩阵经过全连接层进行线性转换，而为了在维度上与 textCNN 得到的特征向量匹配，本文将全连接层改为平均池化（Average-Pooling）操作，对矩阵$M^{(i)}$进行降维操作，使其维度与$C^{(i)}$一致，因此需要对卷积核的数量作如式 4.16 限制，其中k为词向量维度。

$$\sum_{m=1}^{s}N_m=2k \tag{4.16}$$

进而采用相加的方式对两个特征向量进行融合，最后再进入分类器层得到最终的预测结果。其中，W_7是$(2k,\ l)$维度的参数矩阵。

$$\hat{y}_i=\sigma\left(\left[\text{AvgPooling}\left(M^{(i)}\right)+C^{(i)}\right]\cdot W_7\right) \tag{4.17}$$

第三节　基于 LSAN-CNN 模型的评论维度识别

将本文提出的 LSAN-CNN 模型应用于 ASAP 数据集的粗粒度评价维度识别任务中，模型的输入部分为上一章由 Word2Vec 预训练得到的词向量，由于词向量维度为 768，输入词向量矩阵的维度为(batch_size, 768, seq_len)，其中 batch_size

和 seq_len 分别为批数据的大小和评论文本序列长度，属于模型的超参数。标签部分，原数据集中的标签包含 18 个细粒度的评价维度，为了对粗粒度的评价维度进行识别，需要对原数据的标签进行处理，分离出粗粒度的评价维度。

一、ASAP 数据集标签处理

ASAP 数据集的细粒度评价维度命名方式遵循“粗粒度维度#细粒度维度”格式，例如“Location#Transportation”，其中位置（Location）为粗粒度维度，而交通是否便利（Transportation）为细粒度维度。为了使得数据能够适应粗粒度评价维度识别任务，对数据集作如下处理：当至少存在一个细粒度维度的情感极性不为“未提及”时，则对应的粗粒度维度赋予“提及”标签，具体样例如图 4-3 所示。

细粒度评价维度	情感极性
Location#Transportation	-2
Location#Downtown	-2
Location#Easy_to_find	-2
Service#Queue	-2
Service#Hospitality	1
Service#Parking	-2
Service#Timely	-2
Price#Level	0
Price#Cost_effective	-2
Price#Discount	0
Ambience#Decoration	1
Ambience#Noise	1
Ambience#Space	1
Ambience#Sanitary	1
Food#Portion	-2
Food#Taste	-2
Food#Appearance	1
Food#Recommend	-2

粗粒度评价维度	是否提及
Location	0
Service	1
Price	1
Ambience	1
Food	1

图 4-3 ASAP 数据集粗粒度评价维度提取示例

如图 4-3 所示，以该条评论对应的标签为例，根据每一细粒度评价维度下的情感极性可以看出，“位置”维度均为提及，“服务”维度下提到了“服务态度”，“价格”维度下提到了“价格水平”和“折扣力度”，“环境”维度下提到了“装饰”、“噪声”、“空间”和“卫生程度”，而“菜品”维度下则提到了“外观”，因此在粗粒度维度层面下，除“位置”维度外，其余维度均提及。

对 ASAP 数据集的粗粒度评价维度作简单的统计分析可得到图 4-4，由图 4-4 可知评论文本中提及最多的粗粒度维度为菜品（Food），其次为服务态度（Service）和价格水平（Price）。

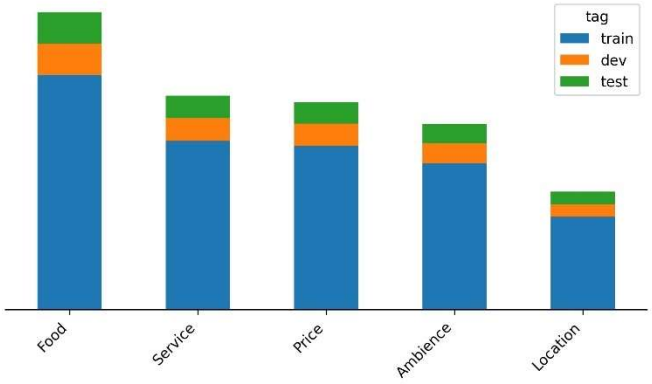

图 4-4　ASAP 数据集粗粒度评价维度分布图

经过上述标签集处理后可以发现，针对 ASAP 数据集的粗粒度评价维度识别任务本质上属于多标签文本分类任务，标签数量为 5 个，分别为“价格”、“位置”、“环境”、“服务”和“菜品”。对应到 LSAN-CNN 模型中，输入的标签词向量矩阵C维度为(batch_size, 768, 5)。

二、模型参数设置

模型的输入部分确定后，还需要初步确定超参数的值。在 LSAN-CNN 模型中，本文对超参数参考原 LSAN 模型作如下设置：①评论文本的最大长度设置为 300；②特征提取层中 Bi-LSTM 网络的隐藏层维度d_h与词向量的维度保持一致，为 768；③Self-Attention 部分系数W_1和W_2的维度d_a为 512；④对于 textCNN 特征提取部分，需要确定的有两部分，一是卷积核的尺寸f_m，二是每个尺寸卷积核的数量N_m。本文将采用四种尺寸的卷积核，高度分别为 2、3、4、5，每种卷积核的数量保持一致，根据式 4.16 的限制，可以推算出每种卷积核的数量为 384 个，因此$f_m=\{2,\ 3,\ 4,\ 5\}$，而$N_m=384$。

模型训练方面，需要确定的超参数如下：①批数据的大小 batch_size，该参数的选取与训练所使用的机器性能有关，batch_size 越大训练时占用的内存就越多，对机器的性能要求就越高；而 batch_size 越小则所需要的训练时间就越长。经过测试，本文将 batch_size 设为 128；②初始学习率大小 learning_rate 和衰减率 decay_rate。模型训练过程中，学习率能够指导模型通过损失函数的梯度调整网络的权重，较小的学习率可以保证参数在更新过程中不会错过局部最优点，但损失函数收敛所耗费的时间也会更长；而较大的学习率会使损失函数难以收敛甚至发散。因此学习率的选取需要同时兼顾训练的时间成本和损失函数的收敛效果，

当前大多数研究人员都会在模型训练中加入学习率衰减策略。本文选取线性衰减策略作为学习率的调整方案，即在训练过程中，学习率从初始学习率开始每隔一段时间按照衰减率线性下降。如此设置的好处是，初始训练阶段损失函数能够在较大学习率下快速向最优点靠近，而训练后期又能在较小学习率下缓慢寻找全局最优点。本文将初始学习率 learning_rate 设置为 0.001，线性衰减率 decay_rate 设置为 0.7；③迭代次数 epochs，该参数决定模型遍历全部样本进行参数迭代更新的次数，epochs 越大，则模型训练所花费的时间就越长，导致过拟合的可能性也越大，而 epochs 越小则模型可能无法收敛。因此，本文对不同 epochs 下模型的收敛情况作了对比后，确定模型的迭代次数为 7 次。

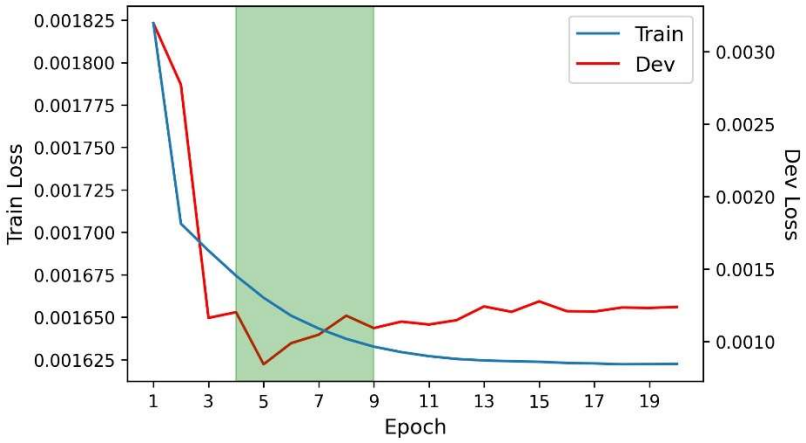

图 4-5　不同 epoch 下 LSAN-CNN 模型损失函数变化图

表 4-1　LSAN-CNN 模型参数设置

参数	参数值
max_seq_len	300
d_h	768
d_a	512
f_m	{2, 3, 4, 5}
N_m	384
batch_size	128
learning_rate / decay_rate	0.001 / 0.7
Epochs	7

第四节　评价指标与结果分析

一、模型评价指标

为了体现本文所提出的 LSAN-CNN 模型的分类效果，并探究本文所提出模

型相较于原模型在分类效果上是否有所提升，需要指定统一的评价指标。对于分类模型，最直观的评价指标为准确率（Accuracy），即完全分类正确的样本数占总样本数的比例。在多分类任务中该评价指标尚且合理，而在多标签分类任务中，由于一个样本可能包含多个标签类别，使用准确率作为评价指标虽然能从整体上反应模型的分类效果，但忽略了模型在单个类别上的分类准确率。

为了更全面地评价模型的分类效果，本文将对每个标签类别计算单独计算评价指标，而后对所有类别的评价指标取平均作为最终的指标计算结果。具体地，本文将采用宏平均精确率（Macro-Precision）、宏平均召回率（Macro-Recall）和宏平均 F1（Macro-F1）作为模型分类效果的综合评价指标。

首先，针对单个类别可以得到如图 4-6 所示的混淆矩阵，其中 a 为模型正确预测属于该类别的样本数，b 为模型无法识别出该类别的样本数，c 为模型误将不属于该类别的样本划分到该类别的样本数，d 为模型正确预测不属于该类别的样本数。由此可以计算出精确率（Precision）、召回率（Recall）和 F1（F1-Score）三个指标，精确率用于反映模型预测为正例的样本中其真实标签类别也为正的样本比例，召回率用于反映真实标签为正的样本中被模型准确识别的样本比例，而 F1 则是精确率和召回率的调和平均数，综合反映模型在该类别上的分类效果。

Prediction / GroundTruth	Positive	Negative
Positive	a	b
Negative	c	d

图 4-6　混淆矩阵示意图

$$\mathrm{Precision}=\frac{a}{a+c} \tag{4.18}$$

$$\mathrm{Recall}=\frac{a}{a+b} \tag{4.19}$$

$$\mathrm{F1}=\frac{2\times\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} \tag{4.20}$$

进而，可利用单个类别的精确率、召回率和 F1 计算宏平均指标，定义为：

$$\mathrm{Macro-Precision}=\frac{1}{\mathrm{n}}\sum_{i=1}^{\mathrm{n}}\mathrm{Precision}_i \tag{4.21}$$

$$\mathrm{Macro-Recall}=\frac{1}{\mathrm{n}}\sum_{i=1}^{\mathrm{n}}\mathrm{Recall}_i \tag{4.22}$$

$$\mathrm{Macro-F1}=\frac{1}{\mathrm{n}}\sum_{i=1}^{\mathrm{n}}\mathrm{F1}_i \tag{4.23}$$

其中，i为标签类别总数，本文研究中$n=5$。

二、模型训练结果

由于本地设备性能有限，本文研究所涉及模型训练部分均在谷歌的 Colab 云计算平台运行，硬件方面 Colab 平台提供了英伟达的 Tesla T4 系列 GPU，拥有 16GB 显存，在上述参数条件下，单个 epoch 耗时 6 分钟左右，7 个 epoch 训练总计耗时约 42 分钟。损失函数在训练集和验证集上的变化趋势如图 4-7 所示。

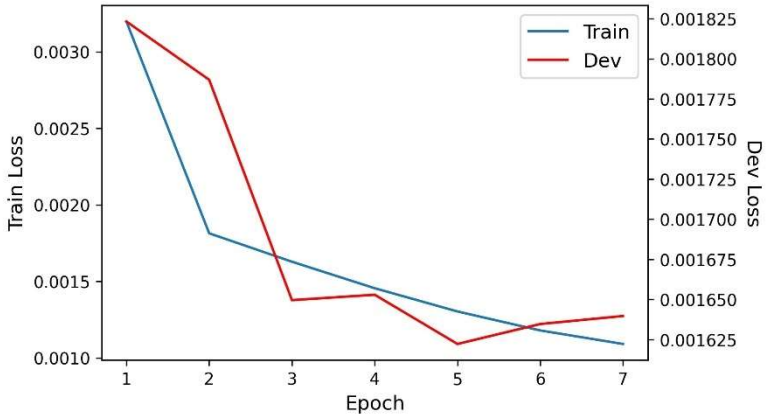

图 4-7　LSAN-CNN 模型训练过程损失函数变化图

根据模型的预测结果计算各评价指标的值得到表 4-2，由表 4-2 可知，无论是对于训练集、验证集还是测试集，三个指标中模型的宏精确率均比宏召回率稍高，在测试集上取得了 93.58%的宏精确率和 91.87%的宏召回率，而宏 F1 为 92.70%。一般而言，模型的精确率和召回率变化会呈反向，提高精确率会使得召回率降低，而提高召回率又会使得损失精确率，因此需要在两个指标之间进行一定的取舍。本文所采用的模型在 9 个 epoch 的训练后精确率比召回率稍高，表明模型偏向于对标签预测的准确性，而可能对部分标签的识别存在遗漏，即对标签宁可漏判也不误判。该结果也符合本文研究的预期，若存在对标签的误判，则会

向后续模型传递冗余信息，直接影响下游基于粗粒度评价维度融合的情感极性分类任务的效果。

表 4-2 LSAN-CNN 模型训练结果

	Train	Dev	Test
Macro-Precision	0.9581	0.9365	0.9358
Macro-Recall	0.9502	0.9189	0.9187
Macro-F1	0.9536	0.9270	0.9270

具体地，由图4-8所示的单个类别下评价指标的值可以发现，无论是精确率、召回率还是 F1-Score，“菜品”（Food）类别的评价指标高，而“位置”（Location）类别最低，表明模型在“菜品”这一粗粒度评价维度的识别效果最好，而对“位置”维度的识别效果较差。而总体上看，除“位置”外，各评价维度的指标值都在 90%以上，表明模型在各个粗粒度评价维度的识别方面表现较好。

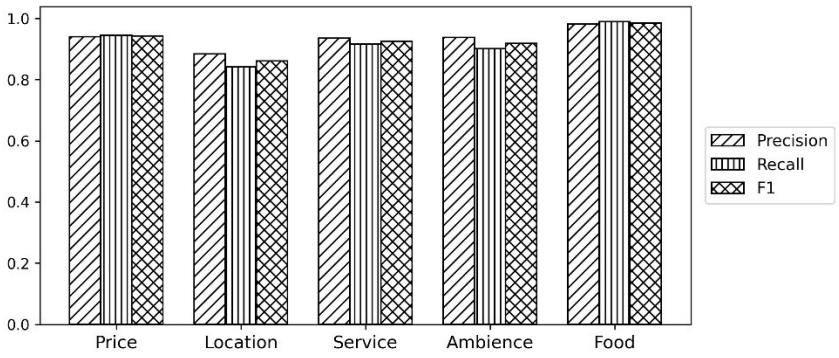

图 4-8 单个粗粒度评价维度下模型的识别效果

LSAN-CNN 模型在测试集的预测实例如图 4-9 所示。第一、二个实例中模型对评论文本所包含的粗粒度评价维度作出了准确识别；第三个实例中，模型只识别出了“服务”和“菜品”维度，无法识别到“价格”维度，主要是因为评论中的“价格”主要体现在对性价比的描述上，语义表达相对比较隐晦，模型无法准确识别；而第四个实例中模型识别到评论文本涉及“服务”维度，而事实上并未提及，模型在该评论文本上出现了误判。

Review	Label	Predicted Label
到顺德，当然要去出名的民信甜品店吃个双皮奶啊，估嘅到民信还进入多元化发展！甜品店做西餐！！因为我们只有2个人，所以点不了那么多，但也点了4个菜啊！呵呵呵！以前在佛山已经吃过它的双皮奶了，很香滑！不过偏甜。这边就要试顺德炸牛奶！果然很不错啊，比我家这边的所谓炸牛奶好吃多了！外脆内滑。苦瓜榨菜剪蛋角也很不错，就是太咸了。最估嘅到就系甜品店出的西餐出品不是一般的好呢！我地入店到座位坐下期间就见到有很多饭包饭出炉！所以我地都点一个试了！嗯，值得点！芝士味嘅会太浓，肉碎与蛋需住炒饭嘅好好味！最紧要就系出品价钱都不贵！性价比很高的一家店！	价格、菜品	价格、菜品
感谢大众点评给我抽中100的代金券，和男票一起去的，从一号线莲花路出来乘公交753到龙茗路顾戴路下车，走过去没多远，点了四个菜，火烧黄牛筋是最喜欢的，锅边贴的是玉米面饼子，酥着汤汁很好吃，牛筋和干笋一起烧很入味，农家手撕豆腐是有汤的，尝了下豆腐很像小时候吃的那种手工豆腐，汤很鲜的说，全部吃完了，重鱼向来是我喜欢的。甜甜脆脆的。最后来了素菜炒菜心，清清肠胃，一起消费166元，吃的很饱，环境还不错，还有免费姜茶喝。愉快的体验。	价格、位置、环境、菜品	价格、位置、环境、菜品
【本格】本格不论是从食材本身还是服务来说，品质都很高。推荐的菜式有：各类刺身（个人最喜欢牡丹虾和扇贝刺身，这和我当天的胃口也有关。刺身切片很厚实，都是非常新鲜的食材，但总觉得那片大腩不够油润。我印象最深刻的是紫玉的金枪鱼大腩寿司，真的入口即化。）；海胆军舰（甜滑。一开始我觉得一口吃不下，就先吸了一口海胆，一下子就滑进了嘴里，没有一丝腥味，微甜）金枪鱼泥萝卜丁卷（我很喜欢，鱼肉当然也是高品质的，和萝卜丁很配）最后推荐当日的甜品南瓜蛋挞，这是唯一一道我想单点的“菜”，不过这是送的……不知道下次去还吃不吃得到\\n之所以这次不是很满意，之所以没给五星，主要是因为搭配的缘故，吉拉多生蚝、爆蝉、三文鱼籽军舰和味增汤都是含盐量较高的食物，真的把我齁住了(·д·) 息，最好不要点厚蛋烧...	价格、服务、菜品	服务、菜品
心爱的宝贝生日，做妈妈的自己早早就物色蛋糕店了，挑来挑去，好不容易挑了这家金之信蛋糕连锁店（墩和），看网上的评价反映不错，提前一天下单，下好单就打电话过去预约第二天自提。昨天下午四点钟左右，就乘坐464路直接到墩和总站下车，位置在电话里提醒挺好的，挺方便，就在附近208车站附近，位置很显眼，不一会儿就看到了。进入店内就在柜台看到自己的下单定的蛋糕，当时一看蛋糕外形很漂亮，有点小错误，就是听名字嘴音写错了，不过服务员很快就拿到制作蛋糕的房间，重新用新的朱古力牌写名字，嗯，字写得很漂亮！很愉快的自提蛋糕过程，提完蛋糕就直接出门旁边坐464路回家了。赞一个！到了晚上，特别邀请了很多小伙伴们过来吃蛋糕，蛋糕的外形很漂亮，个个小伙伴都很喜欢，个个迫不及待地想吹蜡烛和切蛋糕了，哈哈，气氛很好，特别开心，终于许完愿吹了蜡烛切好蛋糕，小伙伴们终于如愿以偿地吃到蛋糕了，“好吃”“我还要一块”“还要”，看看，一不会儿，一个大蛋糕地就剩下一小块了，不错，过了一个很开心的生日，祝愿小宝贝健康成长，天天开心！下次还再来尝这里的蛋糕吧。	位置、菜品	位置、服务、菜品

图 4-9 ASAP 测试集粗粒度评价维度预测示例

三、模型对比分析

为验证本文所提出的 LSAN-CNN 模型的有效性，将 LSAN-CNN 的结果与 LSAN 模型进行比较。具体地，利用原 LSAN 模型在 ASAP 数据集上进行训练，参数设置与 LSAN-CNN 模型的训练过程保持一致，经过 9 个 epoch 的训练后对 ASAP 测试集的粗粒度评价维度进行预测，得到表 4-3 和 4-4 所示的结果。由表 4-3 可知，相比 LSAN 模型，LSAN-CNN 模型在 ASAP 数据集的粗粒度评价维度识别任务上表现更佳，模型的宏精确率、宏召回率和宏 F1 分别较 LSAN 模型提升了 1.92%、0.87%和 1.47%。而从各个维度的评价指标上看，在“价格”、“位置”和“环境”维度的识别中，LSAN-CNN 模型无论是在精确率、召回率还是 F1 上均优于 LSAN 模型；在“服务”维度的识别中，LSAN-CNN 模型的召回率相较于 LSAN 模型要低，也导致了模型的 F1 较低，表明 LSAN-CNN 在该维度的识别上与 LSAN 模型相比存在一定程度的漏判；在“菜品”维度的识别中，LSAN-CNN 模型的精确率低于 LSAN 模型，表明模型在该维度的识别上与 LSAN 模型相比存在一定程度的误判。

表 4-3 LSAN-CNN 模型和 LSAN 模型预测效果对比

	LSAN-CNN	LSAN
Macro-Precision	**0.9358**	0.9166
Macro-Recall	**0.9187**	0.9100
Macro-F1	**0.9270**	0.9123

表 4-4 LSAN-CNN 模型和 LSAN 模型单个评价维度预测效果对比

	Precision		Recall		F1	
	LSAN-CNN	**LSAN**	**LSAN-CNN**	**LSAN**	**LSAN-CNN**	**LSAN**
Price	**0.9401**	0.9245	**0.9448**	0.9298	**0.9422**	0.9259
Location	**0.8844**	0.8714	**0.8415**	0.8407	**0.8609**	0.8537
Service	**0.9354**	0.8806	0.9163	**0.9318**	0.9255	**0.9310**
Ambience	**0.9377**	0.9231	**0.9012**	0.8639	**0.9186**	0.8669
Food	0.9812	**0.9834**	**0.9897**	0.9847	**0.9854**	0.9840

第五节 小结

本章介绍了在多标签文本分类领域应用效果较好的 LSAN 模型的基本原理，在此基础上对模型存在的不足加以改进，在特征提取层引入 textCNN 捕捉文本的 n-gram 特征，提出了 LSAN-CNN 模型。利用该模型对 ASAP 数据集的“价格”、“位置”、“服务”、“环境”和“菜品”五个粗粒度评价维度进行识别，并使用宏精确率、宏召回率和宏 F1 作为衡量模型分类效果的评价指标。经过 7 个 epoch 的训练，LSAN-CNN 模型在 ASAP 测试集上取得 93.58%的宏精确率、91.87%的宏召回率和 92.70%的宏 F1，相较于 LSAN 模型分别提升 1.92%、0.87%和 1.47%，表明 textCNN 模块的引入能够更全面地捕捉文本的语义特征，LSAN-CNN 模型相比 LSAN 模型在评价维度识别任务上具有更好的效果。

第五章 基于 ERNIE 的方面级情感极性分类

ASAP 数据集共包含 18 个细粒度评价维度，例如“位置”这一粗粒度维度下又包含“交通是否便利”、“距离商圈远近”和“是否容易寻找”三个细粒度维度，每个细粒度维度下又包含消极、中性、积极和未提及四类情感极性，而本文研究的最终目的是对单条评论文本的每一个细粒度评价维度进行情感极性分类。该任务本质上属于方面级情感分析任务，当前该领域主流的研究方法都是基于预训练模型，利用预训练过程中模型在大规模语料上习得的通用知识，辅以下游任务数据进行微调，便能够取得不错的分类效果。本文研究所采用的情感极性分类模型是百度的 ERNIE 预训练模型，在该模型基础上提出基于字词相似度加权的词向量构建方法，并将上一步中识别到的粗粒度评价维度的词向量与评论文本的词向量基于 Attention 机制进行交互，进而与评论文本的语义特征进行融合，完

成基于 ERNIE 的细粒度评价维度情感极性分类任务。

本章首先对 ERNIE 预训练模型的基本原理作详细阐述，并提出基于字词相似度加权的词向量构建方法，将 ERNIE 模型产生的字向量转换为词向量。其次，为了捕捉评论文本与粗粒度评价维度的关联信息，本章还提出了基于 Attention 机制构造评价维度词向量的方法。进而将评价维度和评论文本的语义特征进行融合，并对比模型改进前后的分类效果，验证本文所提出方法的有效性。

第一节　ERNIE 预训练模型分析

一、预训练模型概述

人类通过不断学习积累更丰富的知识，与人类的认知过程类似，要使得模型在各种任务上取得更好的效果就需要在规模足够大的数据集上进行训练。而数据规模越大，模型训练的时间成本和对硬件资源的要求就会越高。而预训练模型在预训练阶段通常会采用规模尽可能大的数据集，学习通用领域的知识，得到预训练的模型参数，而后只需要根据需求在特定领域数据集上对参数进行微调便可在下游相关任务上取得不错的效果，极大程度上降低了模型训练的时间成本和硬件开销，提高了模型开发的效率。

二、ERNIE 预训练模型

2018 年，Google 的 Devlin 等提出了 BERT 模型[21]，该模型采用堆叠的多层双向 Transformer 架构作为编码器，并在大规模语料上进行预训练，在文本分类、智能问答、文本推断等 11 个自然语言处理任务中实现 SOTA 效果。

表 5-1　BERT 模型参数规模和模型结构

模型	参数规模	模型架构
BERT-Base	110M	12-layer，768-hidden，12-heads
BERT-Large	340M	24-layer，1024-hidden，16-heads

BERT 的预训练任务为 MLM（Masked Language Model）和 NSP（Next Sentence Prediction）任务，在 MLM 任务中，模型随机输入语句的 15%个单词作 Mask 操作，策略如下：①以 80%的概率将单词替换为“[MASK]”标识符；②以 10%的概率替换为随机选取的单词；③以 10%的概率保持不变。该 Mask 策略相对简单，并没有充分利用到输入文本中蕴含的先验信息，因此百度的 Sun 等人在 2019 年提出了 ERNIE（Enhanced Representation through Knowledge Integration）模型[40]。

ERNIE 在 BERT 的基础上进行改进，模型主体结构不变，Base 版本的模型依然由 12 个 Transformer 编码器构成，但在 MLM 预训练任务的 Mask 策略上作了优化，提出了三个层次的 Mask 策略，分别为原有的随机 Mask 策略、实体层面（Entity-level）的 Mask 策略和短语层面（Phrase-level）的 Mask 策略。其中，实体层面的 Mask 主要是对输入文本中提及的人物、地点、机构等实体作随机 Mask 操作，而短语层面的 Mask 则对文本中包含的固定短语作随机 Mask 操作。相较于 BERT，ERNIE 的 Mask 策略能够在 MLM 预训练任务中更完整地捕捉到文本中蕴含的实体和短语相关的先验知识，进而将这些信息整合到单词的语义表示中，使得模型最终得到的词向量包含更丰富的语义信息。

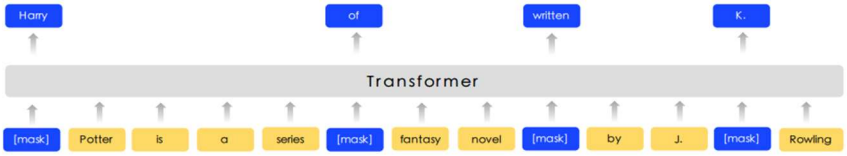

图 5-1　ERNIE 模型 MLM 预训练任务[40]

图 5-2　ERNIE 模型 MLM 预训练任务 Mask 策略[40]

此外，作为国内最大的中文搜索引擎，百度在中文文本数据资源上具有天然优势。 在 ERNIE 的预训练阶段相比 BERT 新增了多个异质性的中文语料，包含中文维基百科、百度百科、百度新闻和百度贴吧等在内的多个中文语料数据集，总规模约为 1.73 亿条。得益于此，ERNIE 能够在预训练阶段从大规模中文语料中学到更多的中文语义信息，也使得 ERNIE 能够应用于中文 NLP 任务中。

表 5-2　ERNIE 模型预训练中文语料

ERNIE 预训练中文语料	
中文维基百科	2100 万
百度百科	5100 万
百度新闻	4700 万
百度贴吧	5400 万

第二节　基于字词相似度加权的词向量构建方法

一、构建方法概述

ERNIE 在预训练阶段从大规模语料上学到的通用知识为模型的参数更新提供了一个很好的基础，只需要在特定任务的语料数据集上加以微调便能取得很好

的效果，这种“大规模预训练+微调”的范式提高了模型的通用性，使得模型能够更好地适应下游不同任务。但是，当前大多数预训练模型包括 ERNIE 在内，其训练得到的语义表示均表现为字向量，即模型对输入的文本作了单字粒度上的切分，得到每一个字对应的语义表示。对于英文版本的预训练模型，其得到的是单词的词向量，而一个单词翻译为中文一般为词语，因此本质上模型学到的是词语的语义信息。但是，中文版本的模型得到的是单个汉字的字向量，对于中文而言，词语所表达的语义信息往往要比单个汉字更加丰富。基于这一前提，本文提出了一种基于字词相似度加权的词向量构建方法。

直观上看，将预训练模型得到的字向量转换为词向量的一种简单的处理方式是直接对字向量进行加总，该方法本质上是对每个字向量赋予了相同的权重，而事实上词语和其对应的字之间的相关性并非完全相同。以词语“三文鱼”及其对应构成的汉字为例，记 $\rho(a, b)$ 为 a 和 b 之间的相关系数，显然有 $\rho(\text{三文鱼}, \text{鱼}) > \rho(\text{三文鱼}, \text{三})$ 以及 $\rho(\text{三文鱼}, \text{鱼}) > \rho(\text{三文鱼}, \text{文})$，因为“三文鱼”属于鱼类，理论上其语义信息与“鱼”字的语义信息最相似。因此简单地对字向量加总为词向量的做法忽略了这种先验信息，得到的词向量并不准确。基于字词相似度加权构建词向量能够考虑到字和词之间的语义相关性，充分捕捉字词之间的关联信息，在词向量的构造方法上比直接加总的方法更合理。

二、字词相似度计算

记词语的词向量为 w，假定该词语由 m 个汉字构成，则 w 对应的字向量为 $\{c_1, c_2, \cdots, c_m\}$。本文采用词向量与字向量的 softmax 归一化向量内积衡量词语与单个汉字之间的关联性，记为 W，具体计算如式，其中词向量 w 和字向量 c_i 均为 k 维的列向量，因此该向量构建方法需要满足词向量与字向量的维度保持一致这一条件。

$$W = softmax(w^T [c_1, c_2, \cdots, c_m]) \tag{5.1}$$

三、加权构建词向量

无论是预训练模型还是词向量模型，其输入部分都需要将文本对齐到同一长度，即需要为模型指定一个代表文本最大长度的超参数。由于对文本的切分粒度不同，同一个文本包含的字符数必然大于或等于其所包含的词语数，因此预训练

模型和词向量模型对最大文本长度的设置也就会存在差异。正因如此，本文所提出的基于字词相似度加权的词向量构建方法在实际计算过程中将会面临以下两种情况：

①词语对应的字在预训练模型中被截断，导致无法匹配或无法完整匹配到对应的字向量。对于这一种情况，本文研究的处理方法是不对字向量作加权，直接取用词向量模型生成的词向量，在后续模型训练中作调整。

②分词后对于单字成词的情况，本文采用的处理方式是取其对应的字向量代替加权词向量进行后续计算。

以图 5-3 中评论文本为例，该评论文本包含 12 个字符，经由预训练模型将产生 12 个字向量$\{c_1, c_2, \cdots, c_{12}\}$。若采用 Python 的 jieba 分词工具，可将其分为 7 个词语，经由词向量模型将产生 7 个词向量$\{w_1, w_2, \cdots, w_7\}$。具体计算方法如式 5.2~5.5：

图 5-3　评论文本对应字向量和词向量

$$W_1 = softmax(w_1^T[c_1, c_2]) \qquad w_1' = [c_1, c_2]W_1^T \tag{5.2}$$

$$W_2 = softmax(w_2^T[c_3, c_4]) \qquad w_2' = [c_3, c_4]W_2^T \tag{5.3}$$

$$W_5 = softmax(w_5^T[c_7, c_8, c_9]) \qquad w_5' = [c_7, c_8, c_9]W_5^T \tag{5.4}$$

$$W_7 = softmax(w_7^T[c_{11}, c_{12}]) \qquad w_7' = [c_{11}, c_{12}]W_7^T \tag{5.5}$$

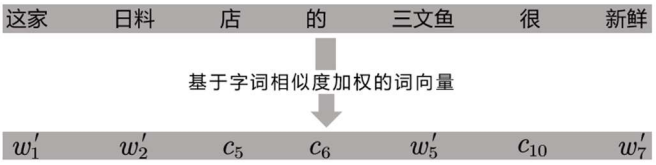

图 5-4　基于字词相似度加权构造的词向量

可以看到，基于字词相似度加权的词向量构造方法将原有的词向量w_i转换为加权的词向量w_i'，其中w_3、w_4和w_6对应的词语属于单字成词，因此使用其对应

的字向量c_5、c_6和c_{10}进行后续计算。

第三节 基于Attention机制的评价维度词向量

一、方法概述

人工对评论文本进行细粒度维度层面的情感分析时，一般可以先判断评论文本所提及的粗粒度维度，再根据评论文本中与粗粒度维度相关的语句的上下文作判断，进一步得到细粒度维度下的情感极性。基于这一思路，为了提取评论文本和粗粒度评价维度之间的关联信息，本文研究提出了基于 Attention 机制构造评价维度词向量的方法。

上一章中，基于 LSAN-CNN 模型可以预测评论文本所提及的粗粒度评价维度，本节将利用 Attention 机制，将该粗粒度评价维度的语义信息与评论文本的语义信息进行交互，得到基于 Attention 机制的评价维度词向量，用于下游计算。记单条评论文本的词向量构成的矩阵为R，维度为$(k,\ n)$；记粗粒度评价维度的词向量矩阵为D，维度为$(k,\ 5)$。其中，k为词向量维度。而评价维度矩阵D的维度为$(k,\ 5)$主要是由于 ASAP 数据集包含 5 个粗粒度评价维度，在评价维度识别任务中，单个评论文本可能涉及一个或多个评价维度，因此在构造评价维度矩阵时需要对维度作补齐操作，不足 5 个粗粒度评价维度的样本将使用“[PAD]”标识符作补充。由于“[PAD]”标识符并没有特殊的语义信息，对其作 Attention 操作也没有意义，因此本文研究在计算过程中加入了 Attention-Mask 操作，即采用一个 0-1 向量表示是否对词语作 Attention 操作，0 表示忽略该位置对应词语的 Attention 操作。具体 Attention 操作计算如式 5.6~5.8：

$$U = tanh(W_a R + b_a) \tag{5.6}$$

$$\alpha = softmax(U^T D) \tag{5.7}$$

$$D' = R\alpha \tag{5.8}$$

其中，W_a和b_a为模型参数，维度为$(k,\ k)$。

二、计算流程

完整的计算流程如图 5-5 所示，首先将预测得到的粗粒度评价维度转换为词向量D并作 padding 操作，同时根据 padding 的情况得到 Attention Mask 向量，

进而利用 Attention 机制提取评论文本和评价维度之间的关联信息，得到基于 Attention 机制的评价维度词向量D'，最后根据 Attention Mask 向量对“[PAD]”标识符的 Attention 操作作掩盖，得到最终的评价维度词向量D''。

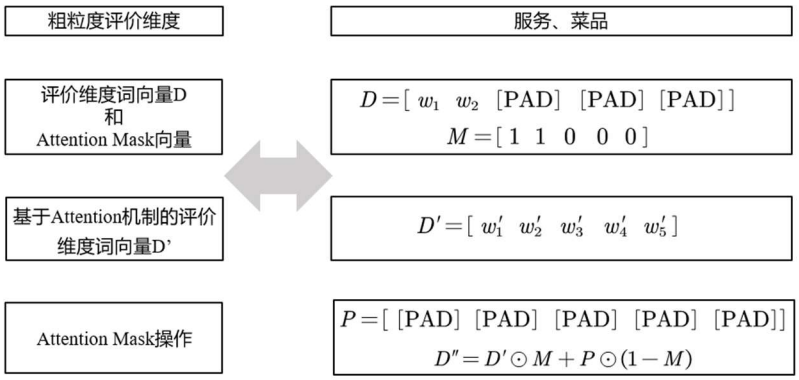

图 5-5　基于 Attention 机制的粗粒度评价维度词向量构建过程

第四节 模型训练

一、情感极性分类模型框架

在对基于字词相似度加权的词向量构建方法和基于 Attention 机制的评论维度词向量的计算方法作详细介绍后，本文研究将对这两种方法所产生的词向量作融合，进一步地提取语义特征，完成细粒度评价维度的情感极性分类任务。完整的任务框架如图 5-7 所示，整个任务流程可以分为三大部分：①利用 ERNIE 预训练模型产生的字向量和 Word2Vec 产生的词向量，基于字词相似度对字向量加权构造新的词向量；②利用注意力机制对第四章中预测得到的粗粒度评价维度的词向量和评论文本的词向量作交互，提取两者之间的关联信息，得到基于注意力机制的粗粒度评价维度词向量；③将基于字词相似度加权构建的评论文本词向量和基于注意力机制的粗粒度评价维度词向量作拼接，将维度信息融合到评论文本中，并采用 Bi-LSTM 网络进一步提取语义特征，然后经过池化层降维，最后再输入分类器中得到不同细粒度评价维度下的情感极性。

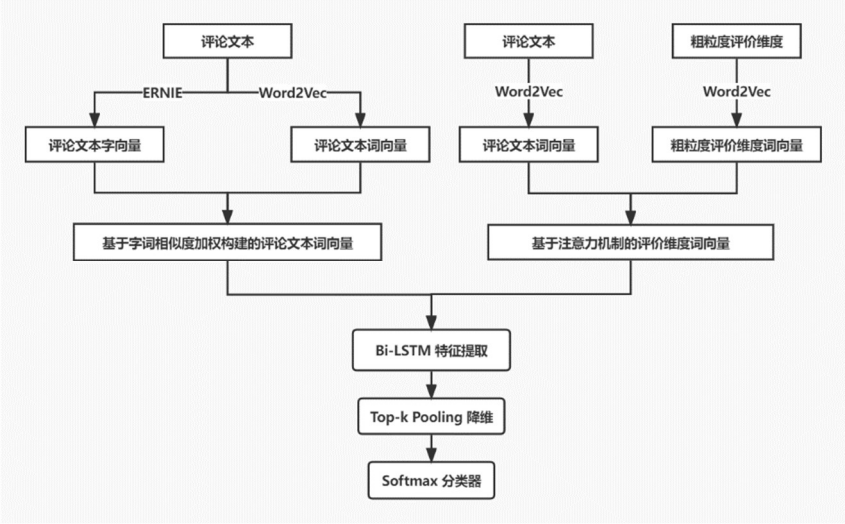

图 5-7　细粒度评价维度情感极性分类任务框架

二、模型损失函数

模型最终将预测不同 18 个细粒度评价维度下的情感极性，而情感极性又包含未提及、消极、中性和积极 4 类，因此对单条评论文本的输出结果将是一个(18, 4)维度的矩阵，每一行代表该维度下各种情感极性的概率。从本质上看，在 softmax 分类层模型同时作了 18 次 4 分类，因此在细粒度评价维度情感极性分类任务上模型的损失函数可以采用交叉熵损失函数。

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{j=1}^{18}\sum_{k=1}^{4} y_{jk}^{(i)} log\hat{y}_{jk}^{(i)} \tag{5.9}$$

三、模型参数设置

模型参数方面，主要有以下几个超参数需要设定：①词向量的维度 k，由于该部分所采用的词向量来自上一个任务中经过 Word2Vec 模型预训练的词向量，且后续语义信息的融合采用的方式是作横向的拼接，词向量的维度应保持一致，为 768；②最大文本长度 max_seq_len，由于对文本的切分粒度不同，该参数在 ERNIE 模型和 Word2Vec 模型中的设置会存在一定的差异，本文将 ERNIE 模型中的最大文本长度保持默认的 512 不变，而对 Word2Vec 模型则仍然采用 300；③Top-k 池化的比例 pool_prop，该参数决定对特征向量的压缩程度，当特征向量为 n 维时，经过 Top-k 池化后将返回前(n×pool_prop)个最大的元素。该比例过大，则起不到特征降维减小计算量的效果，而比例过小，又会损失过多特征向量

的信息，因此本文研究将 pool_prop 参数设置为 0.5。

模型训练方面，相关的参数设置如下：①学习率 learning_rate，本文采用 Adam 优化器，结合学习率线性衰减策略对模型作梯度更新。在优化器中指定 learning_rate 为 0.0005，线性衰减策略中指定预热比率 warmup_rate 为 0.2。设模型总迭代次数为 num_training_steps，则在该线性衰减策略下学习率将先从 0 开始经过(num_training_steps×0.2)步线性增加至 0.0005，随后开始线性下降至 0。使用该策略的好处在于，在模型训练初期能够以较小的学习率保证训练的稳定性，而后期又能以较大的学习率加快损失函数收敛的速度；②批数据大小 batch_size，由于使用了 ERNIE 预训练模型，参数量达到了 110M，batch_size 设置为较小的 16，防止训练过程中内存溢出；③迭代次数 epochs，根据损失函数的变化，将迭代次数设置为。

表 5-3　模型参数设置

参数	参数值
k	768
max_seq_len	512（ERNIE） / 300（Word2Vec）
pool_prop	0.5
learning_rate	0.0005
warmup_rate	0.2
batch_size	16
epochs	20

四、模型输入数据结构

模型输入方面，针对基于字词相似度加权构建词向量模块，为了使得词向量能够与对应的字向量作相似度计算，需要对该模块输入数据作一定的处理。本文研究将输入数据构造为如图 5-6 形式，针对每条评论文本构造索引矩阵，矩阵维度为 n 行 3 列，n 为文本长度。索引矩阵的每一行包含三个元素，第一个元素为词语在词典中的索引，后两个元素为词语对应的字在评论文本中的起始位置和终止位置索引。

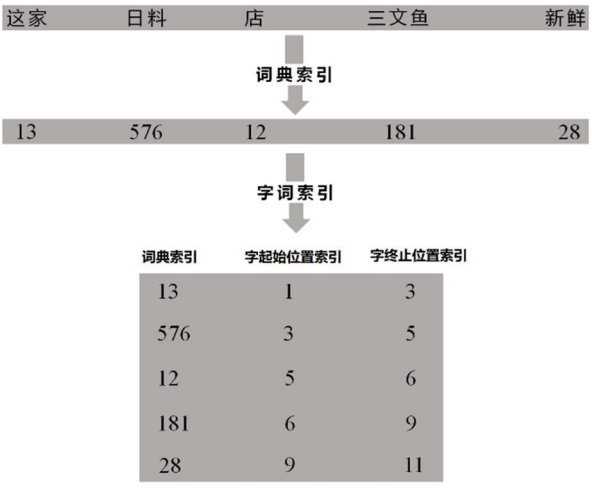

图 5-6 模型输入数据结构

由于 ERNIE 预训练模型包含 12 个 Transformer 编码器，结构较为庞大，参数量较多，若将所有编码器层进行参数更新，则对内存的要求比较高。本文研究虽借助 Google 的 Colab 云计算平台进行模型训练，但内存资源仍然有限，因此本文研究将对 ERNIE 预训练模型的前 11 个 Transformer 编码器的参数进行冻结，训练过程只更新最后一个编码器以及后续层的参数，提高训练效率。

第五节 模型结果分析

一、模型评价指标

由于 ASAP 数据集的标签共包含 18 个细粒度评价维度，而每个评价维度下又包含 4 种情感极性，对评论文本细粒度评价维度的情感极性分类本质上是同时进行 18 个 4 分类任务。因此，模型可以采用与粗粒度评价维度识别任务一致的评价指标，即先计算每个细粒度评价维度的宏评价指标，再对所有评价维度的宏评价指标取平均作为最终衡量模型情感极性分类效果的指标。

$$\mathrm{Macro-Precision}=\sum_{i=1}^{18}\mathrm{P}_{\mathrm{macro}}^{(\mathrm{i})} \tag{5.10}$$

$$\mathrm{Macro-Recall}=\sum_{i=1}^{18}\mathrm{R}_{\mathrm{macro}}^{(\mathrm{i})} \tag{5.11}$$

$$\mathrm{Macro-F1}=\sum_{i=1}^{18}\mathrm{F1}_{\mathrm{macro}}^{(\mathrm{i})} \tag{5.12}$$

二、模型训练结果

由于 ERNIE 预训练模型参数规模较大，本地设备性能有限，模型训练仍在 Google 的 Colab 云计算平台上进行。在单块 Nvidia Tesla T4 显卡条件下，模型训练单个 epoch 耗时约 2 小时，20 个 epoch 总训练耗时约为 40 个小时。训练过程中，模型损失函数的变化过程如图 5-8 所示，经过 20 个 epoch 的迭代，损失函数趋于平稳。训练结束后，将所得模型在测试集语料上做预测，并计算预测结果的宏精确率、宏召回率和宏 F1，得到如表 5-4 结果。

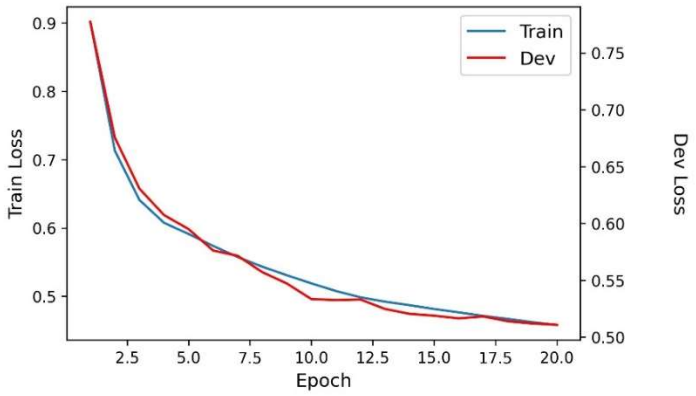

图 5-8　模型训练过程中损失函数变化图

表 5-4　细粒度评价维度情感极性分类结果

评价指标	指标值
Macro-Precision	0.6975
Macro-Recall	0.7003
Macro-F1	0.6989

在细粒度评价维度的情感极性分类任务中，模型在测试集上的宏精确率为 0.6975，宏召回率为 0.7003，宏 F1 为 0.6989。从指标值上看，模型的分类效果并没有很好，究其原因，本文总结为以下两个方面：①模型所要完成的任务是需要同时判断 18 个细粒度评价维度下的情感极性，而每个评价维度下的情感极性又有 4 种可能，相比于简单的二分类或多分类，同时处理 18 个 4 分类任务对模型的挑战更大。因此，要想在评价指标上取得 0.9 以上的表现是比较困难的；②该任务中将上一阶段预测所得粗粒度评价维度信息与评论文本作了 Attention 交互，并作了特征融合，因此上一阶段粗粒度评价维度识别任务的识别效果会直接影响下游情感极性分类的结果，换言之，误差的传导会在一定程度上影响到该任务模型的分类表现。但总体上看，将基于字词相似度加权的词向量构建方法引入

到 ERNIE 预训练模型中，在 ASAP 数据集的情感极性分类上取得的评价指标值都在 0.65 以上，而宏召回率则突破了 0.7，模型分类效果在可接受范围内。

具体地，使用该模型对测试集评论文本的预测实例如图 5-9 所示。

Review	Coarse-grained Dimension	Predicted Label
到顺德，当然要去出名的民信甜品店吃个双皮奶啊，估唔到民信还进入多元化发展！甜品店做西餐！！因为我们只有2个人，所以点不了那么多，但也点了4个菜啊！呵呵呵！以前在佛山已经吃过它的双皮奶了，很香滑！不过偏甜。这边就要试顺德炸牛奶！果然很不错啊，比我家这边的所谓炸牛奶好吃多了！外脆内滑。苦瓜榨菜煎蛋角也很不错，就是太咸了。最估唔到就系甜品店出的西餐出品不是一般的好呢！我地入店到座位坐下期间就见到有很多饭包饭出炉！所以我地都点一个试了！嗯，值得点！芝士味香会太浓，肉碎与蛋混住炒饭咁好好味！最紧要就系出品价钱都不贵！性价比很高的一家店！	价格、菜品	价格 # 价格水平 —— 积极 价格 # 性价比 —— 积极 菜品 # 口味 —— 中性
感谢大众点评给我抽中100的代金券，和男票一起去的，从一号线莲花路出来乘公交753到龙茗路顾戴路下车，走过去没多远。点了四个菜，火烧黄牛筋是最喜欢的，锅边贴的是玉米面饼子，酥香汤汁很好吃。牛筋和千笋一起烧很入味，农家手撕豆腐是有汤的，尝了下豆腐很像小时候吃的那种手工豆腐，汤很鲜的说，全部吃完了。重鱼由来是我喜欢的。甜甜脆脆的。最后来了素菜炒菜心，清清肠胃，一起消费166元，吃的很饱，环境还不错，还有免费姜茶喝。愉快的体验。	价格、位置、环境、菜品	价格 # 价格水平 —— 中性 价格 # 折扣力度 —— 积极 位置 # 交通便利程度 —— 积极 环境 # 装修情况 —— 积极 环境 # 嘈杂程度 —— 积极 环境 # 就餐空间 —— 积极 环境 # 卫生情况 —— 积极 菜品 # 口味 —— 积极

图 5-9　ASAP 测试集细粒度评价维度情感分析效果

三、消融实验

为进一步探究本文所提出的基于字词相似度加权的词向量构建方法和基于 Attention 机制的评价维度词向量在模型中的有效性，本文还设计了如下消融实验：①删除模型架构中的字词相似度加权构建词向量的模块（w/o weighted embedding），直接采用预训练模型生成的字向量与基于 Attention 机制的评价维度融合进行后续计算，该实验主要验证字词相似度加权构建词向量模块的有效性；②删除基于 Attention 机制构建评价维度词向量模块（w/o dimension embedding），直接将加权构建的词向量输入后续网络层进行计算，该实验主要研究基于 Attention 机制的评价维度词向量对模型分类效果的影响。

保持模型其余参数设置不变，同样基于 Colab 平台进行模型训练，得到实验结果如表 5-5 和图 5-10 所示。

表 5-5　消融实验结果

	w/o weighted embedding	w/o dimension embedding	ours
Macro-Precision	0.6698	0.6780	**0.6975**
Macro-Recall	0.6715	0. 6883	**0.7003**
Macro-F1	0.6707	0. 6831	**0.6989**

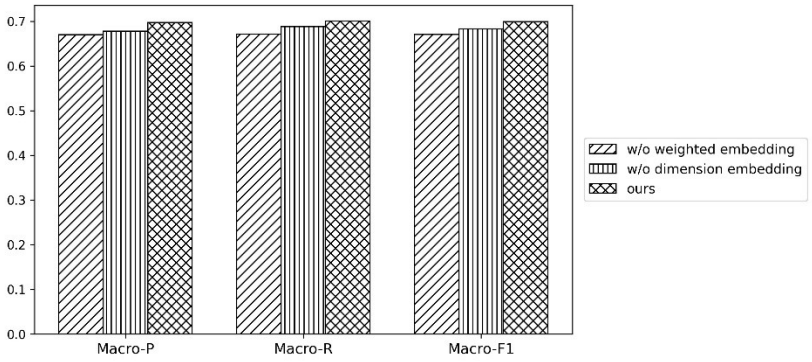

图 5-10　消融实验结果

由消融实验结果可以得出以下结论：①与删除字词相似度加权构建词向量模块的方法相比，本文所采用的模型在宏精确率、宏召回率和宏 F1 上分别提升 2.77%、2.88%和 2.82%，表明基于字词相似度加权构建词向量与直接采用 ERNIE 预训练模型的字向量建模相比，能够包含更丰富的语义信息，在 ASAP 数据集上能够取得更好的情感极性分类效果；②与删除基于 Attention 机制构建评价维度词向量模块的模型相比，本文所采用的模型在宏精确率、宏召回率和宏 F1 上分别提升 1.95%、1.2%和 1.58%，表明基于 Attention 机制构造评价维度词向量能够很好地捕捉评价维度与评论文本之间的关联信息，对评价维度与评论文本的语义信息融合能够进一步提升模型的情感极性分类表现。

第六节 小结

本章介绍了 ERNIE 预训练模型的基本原理，针对中文领域预训练模型采用字向量建模存在的不足，提出了基于字词相似度加权的词向量构建方法。在此基础上，为捕捉粗粒度评价维度与评论文本之间的关联语义信息，本文提出了基于 Attention 机制构建评价维度词向量的方法，对评价维度和评论文本的词向量作 Attention 交互。最终将基于字词相似度加权的评论文本词向量和基于 Attention 机制的粗粒度评价维度词向量融合，输入后续特征提取器和分类器中，得到不同细粒度评价维度下的情感极性。实验结果显示，本文所提出的模型改进方法在细粒度评价维度情感极性分类任务中的宏精确率为 0.6975，宏召回率为 0.7003，宏 F1 为 0.6989。为进一步探究所提出的改进方法的有效性，本文还做了对应的消融实验，结果显示，基于字词相似度加权的词向量构建方法使得模型在宏精确率、宏召回率和宏 F1 上分别提升 2.77%、2.88%和 2.82%，而基于 Attention 机制构建评价维度词向量进而与评论文本做语义特征融合的策略使得模型在三个评价指

标上分别提升 1.95%、1.2%和 1.58%。总体上看，消融实验结果表明本文所提出的基于字词相似度加权的词向量构建方法是有效的。

第六章 总结与展望

第一节 研究总结

本文以 ASAP 餐饮消费在线评论数据集为研究对象，将方面级情感分析任务分为两阶段完成。第一阶段对数据集中的粗粒度评价维度进行识别，第二阶段将第一阶段预测得到的粗粒度评价维度的语义信息融入到评论文本中，完成细粒度评价维度下的情感极性分类。针对该研究，本文总结如下：

①文本向量化表示方面，本文选用 Word2Vec 模型作为词向量模型。本文先对原始评论文本进行简繁转换、去除标点符号和乱码、分词、去停用词等预处理，进而将处理好的语料作为 Word2Vec 模型的输入。经过模型的无监督训练得到预训练的词向量，并通过词向量相似度与对应词语的语义相似度的匹配程度，判断词向量模型的训练效果。研究结果表明，Word2Vec 模型得到的预训练词向量能够包含较为准确的语义信息，模型训练效果较好。

②在粗粒度评价维度识别任务上，本文选取多标签文本分类领域应用效果较好的 LSAN 模型作为基准模型，针对其在特征提取层上只采用 Bi-LSTM 网络的缺陷，引入 textCNN 特征提取器，提出了 LSAN-CNN 模型。textCNN 模块独有的卷积结构使其能够提取文本的局部语义信息，结合 Bi-LSTM 模型所提取文本的时序信息结合，能够得到更完整的语义特征。将 LSAN-CNN 模型在 ASAP 语料上进行训练，并采用宏精确率、宏召回率和宏 F1 作为模型评价指标，在测试集上的预测结果显示 LSAN-CNN 模型的宏精确率为 0.9358，宏召回率为 0.9187，宏 F1 为 0.9270。为验证本文引入的 textCNN 特征提取器的有效性，将 LSAN-CNN 模型与 LSAN 模型的效果作比较，结果显示，LSAN-CNN 模型相较于 LSAN 模型，在宏精确率、宏召回率和宏 F1 的表现上分别提升了 1.92%、0.87%和 1.47%，表明本文对 LSAN 模型所作改进是有效的，textCNN 能够在一定程度上增强特征提取能力。

③在细粒度评价维度情感极性分类任务上，本文选取了 ERNIE 预训练模型作为底层结构，但中文 ERNIE 模型生成的是字向量，而中文词语所表达的语义信息往往比单个汉字更为丰富，因此本文提出了基于字词相似度加权的词向量构

建方法。该方法以 ERNIE 预训练模型产生的字向量为基础，利用传统词向量模型 Word2Vec 产生的词向量与组成该词语的汉字的字向量计算相似度，再加权求和构造词向量。此外，为充分利用评论文本中所提及的粗粒度评价维度的信息，本文还将第一阶段中基于 LSAN-CNN 模型预测所得的粗粒度评价维度的词向量与评论文本做 Attention 交互，构造基于 Attention 机制的评价维度词向量，并于评论文本的加权词向量做拼接融合操作。实验结果表明，本文所采用的情感极性分类模型在 ASAP 测试集上取得 0.6975 的宏精确率、0.7003 的宏召回率和 0.6989 的宏 F1。为进一步探究所提出的改进方法的有效性，本文还做了相关消融实验，结果显示，基于字词相似度加权的词向量构建方法使得模型的宏精确率、宏召回率和宏 F1 分别提升 2.77%、2.88%和 2.82%，而基于 Attention 机制构建的评价维度词向量使得模型在三个评价指标上分别提升 1.95%、1.2%和 1.58%，消融实验结果表明本文所提出的改进方法是有效的。

第二节 研究展望

本文研究仍然存在进一步优化的方向：

①本文研究所采用的数据来源于美团开源的 ASAP 数据集，虽然数据集经过多重人工标注，质量较高，但在标签均衡性上仍然存在缺陷，即涉及某些评价维度的样本数相对较少。数据不均衡将导致模型对少数样本对应的的评价维度识别存在困难，因此后续的研究可以在数据预处理部分进行数据增强，通过同义词替换、随机插入、回译法等增加少数类样本的数量。

②本文研究将方面级情感分析任务分为两阶段完成，第一阶段先对评论文本所提及的粗粒度维度进行识别，第二阶段将上一步中所识别的粗粒度评价维度融合到评论文本的语义信息中完成细粒度评价维度下的情感极性分类。虽然本文研究所提出的理论方法较改进前模型的分类效果均有所提升，但误差传导仍然是两阶段方法中不可避免的问题，即第一阶段模型的误差会传导到第二节阶段的模型中。在今后的研究中，可以考虑多任务模型，通过构建单个模型同时完成粗粒度维度识别任务和细粒度情感极性分类任务。

参 考 文 献

[1] MARON M E. Automatic indexing: an experimental inquiry[J]. Journal of the ACM (JACM), 1961, 8(3): 404-17.

[2] Boutell M R, Luo J, Shen X, et al. Learning multi-label scene classification[J]. Pattern recognition, 2004, 37(9): 1757-1771.

[3] Tsoumakas G, Katakis I. Multi-label classification: An overview[J]. International Journal of Data Warehousing and Mining (IJDWM), 2007, 3(3): 1-13.

[4] Read J, Pfahringer B, Holmes G, et al. Classifier chains for multi-label classification[J]. Machine learning, 2011, 85(3): 333-359.

[5] Clare A, King R D. Knowledge discovery in multi-label phenotype data[C]//European conference on principles of data mining and knowledge discovery. Springer, Berlin, Heidelberg, 2001: 42-53.

[6] Elisseeff A, Weston J. A kernel method for multi-labelled classification[J]. Advances in neural information processing systems, 2001, 14: 681-687.

[7] Zhang M L, Zhou Z H. ML-KNN: A lazy learning approach to multi-label learning[J]. Pattern recognition, 2007, 40(7): 2038-2048.

[8] Zhang M L, Zhou Z H. Multilabel neural networks with applications to functional genomics and text categorization[J]. IEEE transactions on Knowledge and Data Engineering, 2006, 18(10): 1338-1351.

[9] Nam J, Kim J, Mencía E L, et al. Large-scale multi-label text classification—revisiting neural networks[C]//Joint european conference on machine learning and knowledge discovery in databases. Springer, Berlin, Heidelberg, 2014: 437-452.

[10] Berger M J. Large scale multi-label text classification with semantic word vectors[J]. Technical report, Stanford University, 2015.

[11] KIM Y. Convolutional Neural Networks for Sentence Classification[J]. arXiv:1408.5882, 2014.

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

[13] Liu J, Chang W C, Wu Y, et al. Deep learning for extreme multi-label text classification[C]//Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval. 2017: 115-124.

[14] Chen G, Ye D, Xing Z, et al. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization[C]//2017 international joint conference on neural networks (IJCNN). IEEE, 2017: 2377-2383.

[15] Nam J, Mencía E L, Kim H J, et al. Maximizing subset accuracy with recurrent neural networks in multi-label classification[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 5419-5429.

[16] YANG P, SUN X, LI W, et al. SGM: Sequence Generation Model for Multi-label Classification, Santa Fe, New Mexico, USA, F aug, [C]//Proceedings of the 27th International Conference on Computational Linguistics. 2018: 3915–3926.

[17] Lin J, Su Q, Yang P, et al. Semantic-unit-based dilated convolution for multi-label text classification[J]. arXiv:1808.08561, 2018.

[18] Xiao L, Huang X, Chen B, et al. Label-specific document representation for multi-label text classification[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019: 466-475.

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

[20] Yarullin R, Serdyukov P. BERT for Sequence-to-Sequence Multi-label Text Classification[J]. AIST, 2020: 187-198.

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

[22] Feldman R. Techniques and applications for sentiment analysis[J]. Communications of the ACM, 2013, 56(4): 82-89.

[23] Hu M, Liu B. Mining and summarizing customer reviews[C]//Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. 2004: 168-177.

[24] Moghaddam S, Ester M. Opinion digger: an unsupervised opinion miner from unstructured product reviews[C]//Proceedings of the 19th ACM international conference on Information and

knowledge management. 2010: 1825-1828.

[25] Cruz F L, Troyano J A, Enríquez F, et al. ‘Long autonomy or long delay?’The importance of domain in opinion mining[J]. Expert Systems with Applications, 2013, 40(8): 3174-3184.

[26] Boiy E, Moens M F. A machine learning approach to sentiment analysis in multilingual Web texts[J]. Information retrieval, 2009, 12(5): 526-558.

[27] Jiang L, Yu M, Zhou M, et al. Target-dependent twitter sentiment classification[C]//Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies. 2011: 151-160.

[28] Wagner J, Arora P, Cortes S, et al. Dcu: Aspect-based polarity classification for semeval task 4[J]. 2014.

[29] Tang D, Qin B, Feng X, et al. Effective LSTMs for target-dependent sentiment classification[J]. arXiv:1512.01100, 2015.

[30] Wang Y, Huang M, Zhu X, et al. Attention-based LSTM for aspect-level sentiment classification[C]//Proceedings of the 2016 conference on empirical methods in natural language processing. 2016: 606-615.

[31] MA D, LI S, ZHANG X, et al. Interactive attention networks for aspect-level sentiment classification[C]//Proceedings of the 26th International Joint Conference on Artificial Intelligence. 2017: 4068–74.

[32] PETERS M E, NEUMANN M, IYYER M, et al. Deep Contextualized Word Representations[C]//Association for Computational Linguistics. 2018: 2227–2237.

[33] SUN C, HUANG L, QIU X. Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence[C]//Association for Computational Linguistics. 2019: 380–385.

[34] Jiang Q, Chen L, Xu R, et al. A challenge dataset and effective models for aspect-based sentiment analysis[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019: 6280-6285.

[35] SABOUR S, FROSST N, HINTON G E. Dynamic routing between capsules[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 3859–69.

[36] BU J, REN L, ZHENG S, et al. ASAP: A Chinese Review Dataset Towards Aspect Category Sentiment Analysis and Rating Prediction[C]//Association for Computational Linguistics. 2021:

2069–2079.

[37] Bengioy Y, Simardy P, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE Trans. Neural Netw, 1994, 5: 157-166.

[38] Hochreiter S, Schmidhuber J. LSTM can solve hard long time lag problems[J]. Advances in neural information processing systems, 1996, 9.

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

[40] Sun Y, Wang S, Li Y, et al. Ernie: Enhanced representation through knowledge integration[J]. arXiv preprint arXiv:1904.09223, 2019.

[41] Chen P, Sun Z, Bing L, et al. Recurrent attention network on memory for aspect sentiment analysis[C]//Proceedings of the 2017 conference on empirical methods in natural language processing. 2017: 452-461.

[42] Hussein D M E D M. A survey on sentiment analysis challenges[J]. Journal of King Saud University-Engineering Sciences, 2018, 30(4): 330-338.Feldman R. Techniques and applications for sentiment analysis[J]. Communications of the ACM, 2013, 56(4): 82-89.

[43] Schouten K, Frasincar F. Survey on aspect-level sentiment analysis[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 28(3): 813-830.

[44] Sentiment analysis leveraging emotions and word embeddings[J]. Maria Giatsoglou,Manolis G. Vozalis,Konstantinos Diamantaras,Athena Vakali,George Sarigiannidis,Konstantinos Ch. Chatzisavvas. Expert Systems With Applications . 2017.

[45] Xue W, Li T. Aspect based sentiment analysis with gated convolutional networks[J]. arXiv preprint arXiv:1805.07043, 2018.

[46] Hoang M, Bihorac O A, Rouces J. Aspect-based sentiment analysis using bert[C]//Proceedings of the 22nd nordic conference on computational linguistics. 2019: 187-196.

[47] Sun K, Zhang R, Mensah S, et al. Aspect-level sentiment analysis via convolution over dependency tree[C]//Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). 2019: 5679-5688.

[48] Phan M H, Ogunbona P O. Modelling context and syntactical features for aspect-based sentiment analysis[C]//Proceedings of the 58th Annual Meeting of the Association for

Computational Linguistics. 2020: 3211-3220.

[49] Wan H, Yang Y, Du J, et al. Target-aspect-sentiment joint detection for aspect-based sentiment analysis[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(05): 9122-9129.

[50] 贾闻俊,张晖,杨春明,赵旭剑,李波. 面向产品属性的用户情感模型[J].计算机应用,2016,36(01):175-180.

[51] 曾锋, 曾碧卿, 韩旭丽, 等. 基于双层注意力循环神经网络的方面级情感分析[J]. 中文信息学报, 2019, 33(6): 108-115.

[52] 国显达,那日萨,崔少泽. 基于 CNN-BiLSTM 的消费者网络评论情感分析[J].系统工程理论与实践,2020,40(03):653-663.

[53] 宋威, 温子健. 基于特征双重蒸馏网络的方面级情感分析[J]. 中文信息学报, 2021, 35(7): 126-133.

[54] 沈超,王安宁,方钊,张强. 基于在线评论数据的产品需求趋势挖掘[J].中国管理科学,2021,29(05):211-220.DOI:10.16381/j.cnki.issn1003-207x.2018.1508.

[55] 程梦,洪宇,尉桢楷,姚建民. 融合情感词交互注意力机制的属性抽取研究[J].中文信息学报,2021,35(10):90-100.

[56] 王光,李鸿宇,邱云飞,郁博文,柳厅文. 基于图卷积记忆网络的方面级情感分类[J].中文信息学报,2021,35(08):98-106.

[57] 黄山成,韩东红,乔百友,吴刚,王国仁. 基于 ERNIE2.0-BiLSTM-Attention 的隐式情感分析方法[J].小型微型计算机系统,2021,42(12):2485-2489.

[58] 徐月梅,施灵雨,蔡连侨. 一种基于情感特征表示的跨语言文本情感分析模型[J].中文信息学报,2022,36(02):129-141.