基于细粒度情感分析和菜品名称主题模型的商家推荐

目录

1 实验环境

- 开发平台：PaddlePaddle
- 开发语言：Python 3.9
- 深度学习平台：Pytorch 1.10.1
- GPU 版本：cuda 11.3

2 实验内容

- 实现词嵌入向量模型
- 实现 text-CNN 模型
- 实现 text-CNN 与 LSTM 的并行模型
- 在 LSTM 的基础上添加多头自注意力机制
- 使用 LDA 和 K-means 完成基于菜品名称的口味聚类
- 利用细粒度情感分析结果和口味聚类结果完成商家打分

3 研究背景与意义

情感分析是自然语言处理领域最重要的一个子任务之一，是指通过文本内容计算出相应的情感极性（正面、负面、中性等）。在当今互联网蓬勃发展的今天，线上平台中用户产生的文本数据量迅猛增长。但是，由于对商品、店铺的评价文本不规则且包含消费者的不同观点，用户越来越不能直接从部分评论中获取自己真正想了解的产品质量。因此，进行细粒度文本情感分析研究，分析如何更精确地推测一条评论的情感极性，可以促进产品属性建立、用户画像生成、品质预测等技术的发展，帮助消费者提供更直接的消费建议。

另一方面，不同的人在不同时间点餐对酸、甜、苦、辣、咸的口味的需求是不同的，如果能够通过一家店铺的菜单所包含的菜品名称直接预测出相应菜品的口味，那么用户就不需要再盲目挑选，而是通过自己的口味偏好与店铺菜单的口味分布的

相似度完成直接推荐。

在如今快节奏的社会，更快、更准确的推送是各个线上平台极力追求的目标。以美团为例，其商家推荐的机制目前还基本停留在利用用户评分、店铺距离、用户历史等信息，缺乏一定的个性化和直接性；而对评论文本的利用上，以提取高频词为主（例如“服务好”、“分量足”等）；而对菜品名称的利用上，以作为搜索关键字为主。这样看来，通过细粒度情感分析和菜单口味聚类来为用户提供更直接、更全面的推荐信息，具有很大的研究价值和实际应用价值。

4 国内外相关研究及其现状

传统的文本情感分析研究主要是对句子级或篇章级文本进行整体分析，获取的结果是文本整体所表达的情感极性，但是通常评价文本并不只是表达一种情感倾向，而是对某一事务事物不同方面在同一文本序列中进行表述，这就需要由细粒度情感分析解决。例如在已知评价方面为“服务”和“菜品”时，评论“这家店的菜不算太好吃，但是服务挺满意的”应该被分析为“服务—正面；菜品—负面”。

对于细粒度情感分析，现有的方法主要集中在深度学习方面。Kim 提出 CNN 能够在文本数据中学习到局部重要特征，而在细粒度情感分析任务中能够不用考虑方面词和情感词所处的位置获取方面词相关的情感极性；Xue 和 Li 提出在 CNN 网络中添加门控单元来控制情感信息的去向，并且发现 CNN 在局部特征提取上具有优势，但是却无法捕捉长距离文本语义信息；Ruder 等人提出了基于方面的 aspect-LSTM，通过分层 BiLSTM 对句子间关系进行关联建模，并通过注意力机制实现多方面情感信息的提取；在 AI-Challenger2018 中文细粒度情感分析赛道中，Google 提出的 BERT 在该赛道上取得冠军，彰显了该模型在该任务上优良的性能。

对于菜品名称的口味聚类，我们提出可以创新性地将酸、甜、苦、辣、咸等口味看做是一种主题，这就用到了 LDA 主题模型。潜在狄利克雷分布 LDA 模型是 Blei 提出的一种对离散数据集(如文档集)建模的概率主题模型。Blei 等人利用 LDA 对文本进行建模，然后将建模后的文本使用支持向量机 SVM 进行分类，在降维幅度达到 99% 的情况下提高了文本分类的准确度；李文波等人提出了附加类别标签的 LDA 模型，通过在传统 LDA 模型中融入文本类别信息，提高了该模型的分类能力，克服了传统 LDA 模型用于分类时强制分配隐含主题的缺陷；Xing 等人将 LDA 模型和语言模型结合，并使用基于聚类的方法提高了检索的召回率。

5 相关工作

5.1 实验内容概述

本次实验设计的商家推荐系统分为两个模块，第一个模块是基于 text-CNN 和 LSTM+多头自注意力机制两条并行的分析网络构成的细粒度情感分析模型；第二个

模块是基于 LDA 主题分析模型+K-means 聚类的菜品名称口味聚类模型。

在应用阶段，该系统的输入为商家评论文本和菜单所包含的菜品名称文本，在经过预训练词嵌入向量模型的转换后由文本向量变为数据向量，然后两个模块分别利用各自模型输出店铺的评论情感评分和口味分布向量，并对店铺进行相关评分。评分方法为加权平均，然后系统根据评分选出 top-k 商家进行推荐。

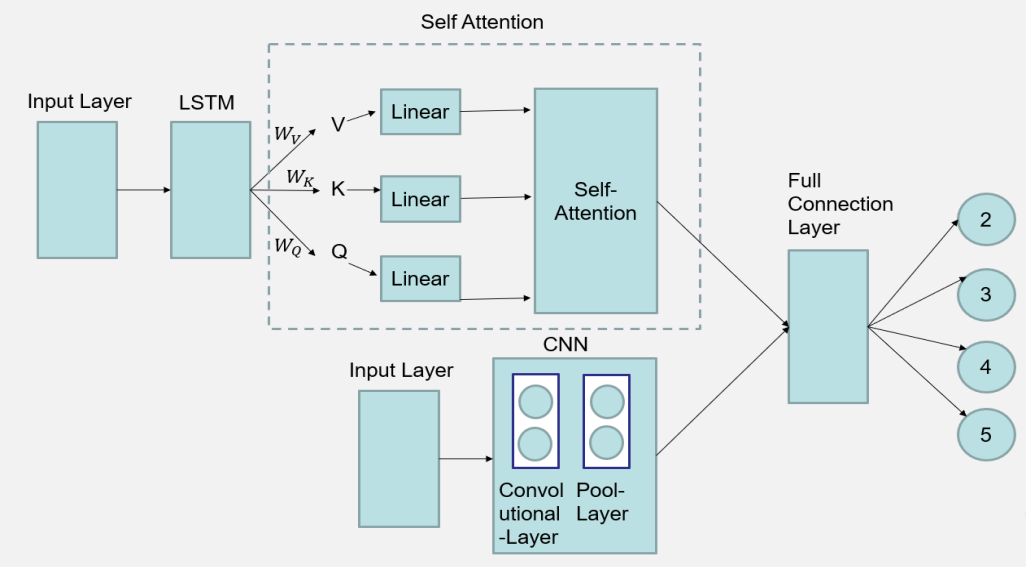

图 5.1.1—细粒度情感分析模型结构示意图

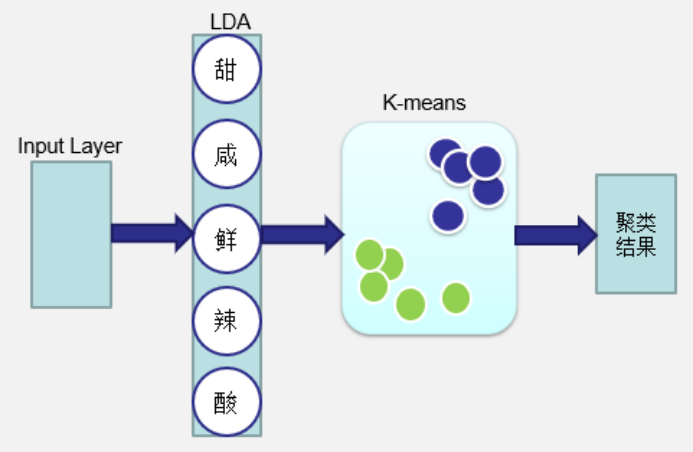

图 5.1.2—菜品名称口味聚类模型结构示意图

5.2 关键问题分析和解决方案

从细粒度情感分析模型提取文本特征角度来说，CNN 通过卷积操作能够提取出评论文本的局部特征（例如修饰关系），能够处理多方面的情感表达信息，但是对全局的情感信息有所欠缺；LSTM 能够识别评论文本的序列特征，因此对全局的情感信息把握得很好，但是对评论文本的局部特征把握不足。为此，我们将这两个模型结合起来形成一个并行结构，利用 text-CNN 模型着重提取文本的局部情感特征，利用 LSTM 模型着重提取文本的全局情感特征，并将它们的输出合并到一个全连接层上，由全连接层计算出分类概率。在这个并行模型的基础上，为了强化 LSTM 划分序列

特征中重要信息的能力，我们在 LSTM 的基础上加入了一个多头自注意力机制。这种方法解决了 LSTM 长距离依赖问题，还提升了 LSTM 特征识别质量。

从细粒度情感分析模型分析速度来说，考虑到我们的应用场景是分析速度要求很高的在线应用情景，模型的分析效率十分关键，这主要约束了模型的复杂度。实践表明，BERT 的计算速率太慢了（分析 15000 条数据需要 1 小时），而如今面向深度学习计算的硬件都是面向 CNN 设计的，且 text-CNN 占有模型的主要计算量，因此我们的这个模型能够保证充足的运算速率（分析 300 条评论只需要约 1 秒）。

对于菜品名称口味聚类来说，一个重要的问题就是 LDA 主题模型只能输出连续性的、目标属于各个类别的概率。应用我们在学习谱聚类时的思想，我们在 LDA 输出类别的基础上再进行一次 K-means，不仅将 LDA 输出结果离散化，还提升了分类精度。这是因为，一些目标在 LDA 中的分析结果可能存在多个类别概率相近的情况，如果直接使用取最大值确定目标分类结果会造成一定的误差。

5.3 基于 text-CNN+LSTM+Multi-Self-Attention 的细粒度情感分析模型

5.3.1 卷积神经网络 CNN

如果使用全连接神经网络处理数据可能存在三个问题：将数据展开为向量会丢失空间信息；参数过多导致训练效率低下，难以调优；参数过多导致模型很可能发生过拟合现象。而卷积神经网络很好地避免了这些缺陷，因为每一层的神经元只与前一层中的一小块区域连接，而不是采取全连接方法。卷积神经网络主要由输入层、卷积层、池化层和全连接层组成。

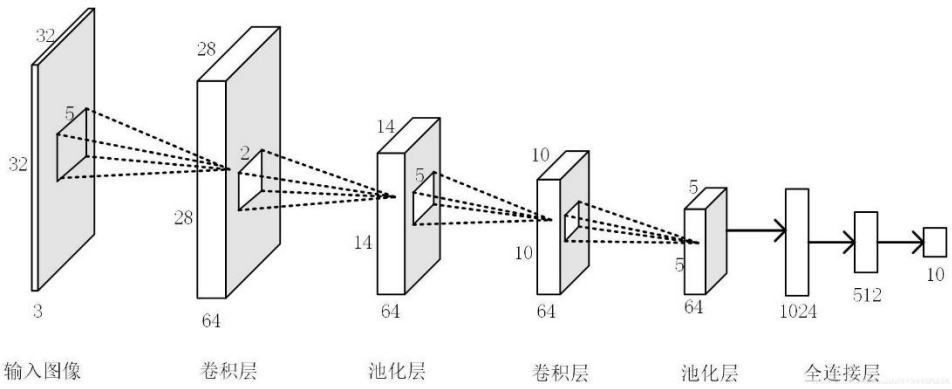

图 5.3.1.1—卷积神经网络结构示意图

卷积层是构建卷积神经网络的核心层，它产生了网络层中大部分得劲计算量。卷积层的参数是由一些可学习的滤波器集合构成的，每个滤波器在深度上和输入数据保持一致，并且在学习过程中滤波器的参数（权值）在数据间是共享的。这样做可以把滤波器的卷积结果看做是一个神经元的输出，有效降低了参数数量。通常在连续的卷积层之间会周期性地插入一个池化层，其作用为逐步降低卷积对象的空间尺寸，进一步减少网络中参数数量，防止过拟合。

5.3.2 面向文本的卷积神经网络 text-CNN

Text-CNN 是 Yoon Kim 等人在 EMNLP 2014 上发布的模型。该模型以预训练的词向量（例如 Glove 和 Word2vec）为输入得到一个嵌入层，这种处理方式使得词向量的维度是固定的（便于处理），而且相对于词的独热表示来说数据量小很多，更重要的是在词嵌入向量空间中语义相近或语法相近的词向量会更加接近。接着是卷积层，相比于一般 CNN 中的卷积核，面向文本的卷积核的长度一般要和词的语法相匹配，例如论文中采用的是 6，表示英文中一般 6 个词能够表达一个相对完整的信息。卷积层的下一层是池化层，采用的是 max-overtime-pooling，即对当前位置所对应的特征图进行最大池化。在池化层的最后加上全连接层和 SoftMax 层进行分类任务。

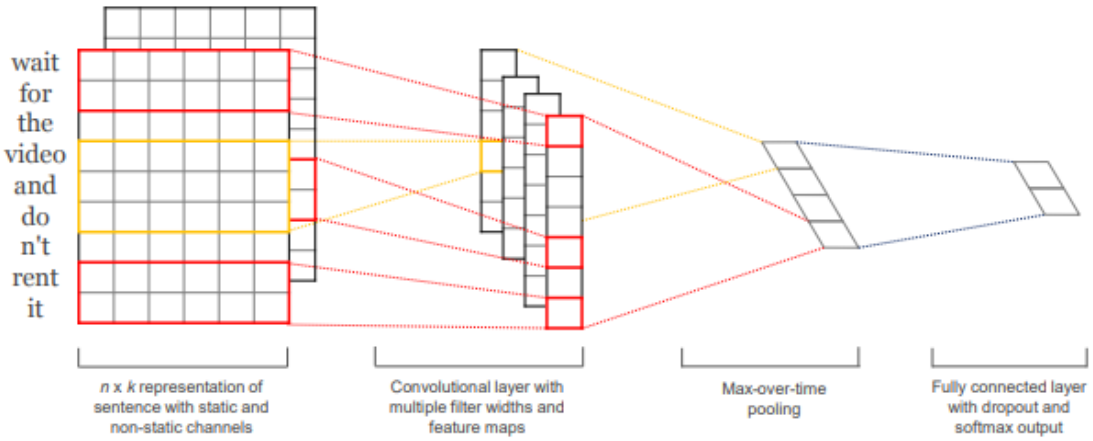

图 5.3.2.1—text-CNN 模型的基本结构

在本实验中我们参考这种模型的结构进行设计。考虑到修饰关系中两个核心词（修饰主题和修饰内容）的距离一般在 2、3、4、5 之间（包含两个词本身）：例如描述菜品味道，可以有“菜/好吃/”、“菜/不/好吃”、“菜/很/不/好吃”，“菜/可能/还算/比较/好吃”等表达方式。我们为此设置四个并行的 text-CNN，令它们的卷积核尺寸分别为 1*2、1*3、1*4、1*5（第一个维度为 1 是因为文本只能单行处理），并将它们最大池化后的结果串联起来传递给后续的全连接层。

5.3.3 长短时记忆神经网络 LSTM

传统神经网络模型结构中上一层神经元的输出是下一层神经元的输入，同一层级的各个神经元是彼此独立的。这对于文本处理任务来说是不现实的，因为自然语言文本的序列特征很强，为此 Elman 等人提出了循环神经网络，其隐层的神经元之间有连接，这样神经元中接收到的数据不仅包含上一时刻隐藏层输出信息，也包含当前输入信息，模型中节点的相连增强了各层之间消息的相互传递，这使得 RNN 模型能够在解决文本序列数据任务上对文本的上下文信息进行有效的综合。

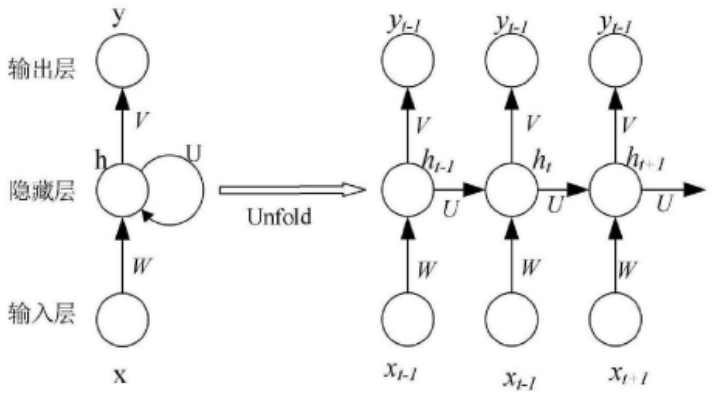

图 5.3.3.1—循环神经网络计算图展开结构

由于 RNN 网络层次的深度随着输入来对网络进行展开，在反向传播时非线性激活函数会由于多个时间步的传播而导致求导后的梯度趋近于无穷或零，因此在计算过程中容易出现梯度爆炸或梯度消失的问题，这就使得 RNN 在输入序列过长时梯度值往往不能传递下去，它就无法有效捕获长距离信息。

于是，LSTM 作为 RNN 的改进结构，包含用来保留历史信息的记忆单元。LSTM 的结构中引入了细胞状态和门结构。其中细胞状态作为网络中信息的传输路径，能够在传递文本信息的同时保留以往时刻的信息并将相关信息传递到后面的记忆细胞中，克服了 RNN 的短时记忆缺点。而门结构是为了控制信息的传递，LSTM 含有输入门、输出门和遗忘门。其中：

- 输入门控制当前时刻的候选状态 c'_t 有多少信息需要保存
- 输出门控制当前时刻的内部状态 c_t 有多少信息需要输出给外部状态
- 遗忘门控制上一时刻的内部状态 c_{t-1} 需要遗忘多少信息

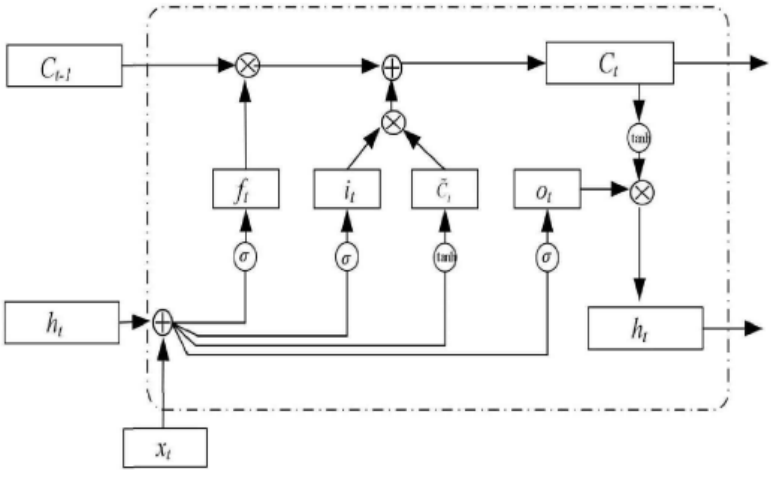

图 5.3.3.2—LSTM 网络模型结构示意图

该模型的计算过程为，首先利用上一时刻的外部状态 h_{t-1} 和当前时刻的输入 x_t，计算出输入门、遗忘门、输出门以及候选状态的值，然后结合遗忘门和输入门来更新内部状态，最后结合输出门将内部状态 c_t 信息传递给外部状态。如果设输入门、遗忘门、输出门分别为 i_t、f_t、o_t，则计算公式为：

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

5.3.4 注意力机制和多头注意力机制

注意力机制 Attention 是一种对文本信息计算注意力权重从而关注到重要信息的组合函数。Bahdanau 等人在处理机器翻译任务时在循环神经网络中增添注意力机制，取得良好成效。

注意力机制主要是与神经网络结合通过分别计算当前输入的不同向量组所对应的权重值，再依据权重值对向量组的向量进行加权平均计算，最终获得注意力向量表示进行输出。如果在 BiLSTM 的基础上添加注意力机制，设第 i 时刻隐藏层输出为 h_t，W_1 和 W_2 表示指定层的权重值，b 为偏置矩阵，则注意力机制的计算公式为：

$$
\begin{aligned}
u_t &= \tanh(W_1 h_t + W_2 b) \\
\alpha_t &= \mathrm{softmax}(\mathrm{u}_t^T, \mathrm{U}_w) \\
\mathrm{v} &= \sum_t \alpha_t h_t
\end{aligned}
$$

传统的注意力机制在文本情感分析任务处理上具有一定局限性，在细粒度情感分析的任务中主要体现为该机制无法将注意力分散到多个评价方面上，导致在细粒度情感极性分析时准确度不足。为此我们进行改进，在 LSTM 上所添加的是多头自注意力机制。这个机制简单来说，就是将模型分为多个头，形成多个子空间，可以让模型取关注不同方面的信息，然后把得到的信息串联起来。

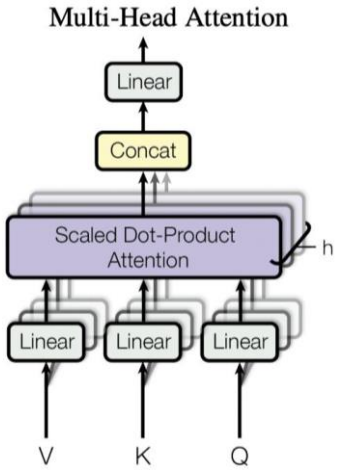

图 5.3.4.1—多头自注意力机制结构示意图

这种注意力机制实质上就是一个寻址过程，通过给定一个任务相关的查询向量 Query，通过计算它与 Key 的注意力分布并附加在 Value 上，从而计算 Attention Value，缓解神经网络的复杂度。其计算公式如下：

$$
\begin{aligned}
&Q_i = QW_i^Q \\
&K_i = KW_i^K \\
&V_i = VW_i^V \\
&head_i = Attention(Q_i, K_i, V_i) \\
&Multi(Q,K,V) = Concat(head_1,...,head_n)W^O
\end{aligned}
$$

在本模型中我们选择该多头注意力机制的头数为 3。

5.4 基于 LDA 主题模型+K_means 聚类的菜品名称口味分类模型

5.4.1 LDA 主题模型

LDA 模型是一种对离散数据集（如文档集）建模的概率主题模型，是一种对文本数据的主题信息进行建模的方法，通过对文档进行一个简短的描述，保留本质的统计信息，有助于高效的处理大规模文档集。它由 3 层生成式贝叶斯网络结构构成，依次为文档集合层、主题层和特征词层。并且该模型基于这样一种前提假设：文档是由若干个隐含主题构成（在本任务中是我们指定的口味类别），而且这些主题是由文本中若干个特定词汇构成，而忽略文档中的句法结构和词语出现的先后顺序。这个特性不会影响到菜品名称分析，因为它只是一个短语，而且词的顺序基本不影响含义的表达。

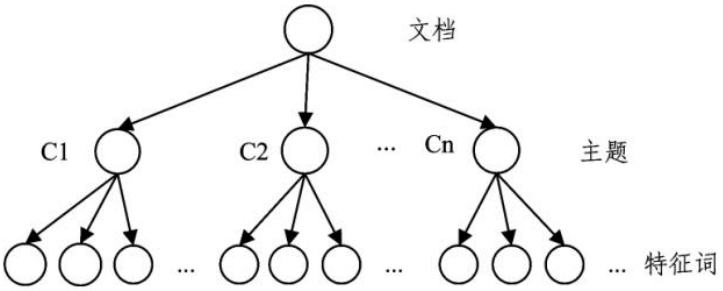

图 5.4.1.1—LDA 主题模型的基本构成

下图展示了 LDA 模型的典型有向概率图，LDA 的初态由参数(α,β)确定，其中α反映了文档集合中隐藏主题之间的相对强弱关系，β反映了所有隐含主题自身的先验概率分布。在模型的计算过程中，θ_k 表示一个文档主题的概率分布，φ_k 表示在特定主题下特征词的概率分布，M 表示文档集的文本数，K 表示文档集的主题数，N 表示每篇文档包含的特征词数。

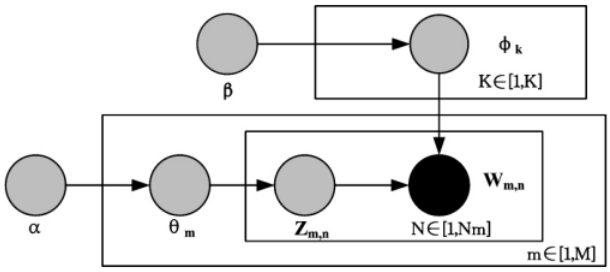

图 5.4.1.2—LDA 有向概率图模型

基于这个有向概率图，可得 LDA 主题概率计算公式为：

$$p(\theta, z, w \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) P(w_n \mid z_n, \beta)$$

因此，LDA 概率主题模型生成文本的过程为：

1）对主题 z 根据狄利克雷分布 Dir(β)得到该主题上一个单词多项式分布向量φ；

2）根据泊松分布 P 得到文本的单词数目 N；

3）根据 Dir(α)得到该文本的一个主题分布概率向量θ；

4）对于该文本 N 个单词中的每一个单词 W_n，从θ的多项式分布 Multinomial(θ)随机选择一个主题 z。然后再从主题 z 的多项式条件概率分布 Multinomial(φ)选择一个单词作为 W_n。

5.4.2 K_means 聚类分析

K-means 算法体现了自顶而下的思想。在本问题我们已经确定了待分类样本的类别数 K（即指定的口味数），下一步进行的是数据的聚类。其具体实现过程为：

输入：训练样本集 D(包含 m 个样本点)和分类数 k。

输出：关于样本点的簇划分 C。

算法：

1 随机初始化 k 个聚类中心。

2 while True：

3 对每个训练样本，打上离它最近的簇中心标签，即 $C^{(t)}(j) = \arg\min_i \| \mu_i - x_j \|^2$ 。

4 对于每一簇，重新计算中心，即 $\mu_i^{(t+1)} = \arg\min_{\mu} \sum_{j:C(j)=i} \| \mu - x_j \|^2$ 。

5 if 样本中心变化量<阈值或 n 个样本的标签不再变化：迭代结束。

K-means 算法和 LDA 模型都有各自的缺陷。传统的 K-means 算法的性能与初始聚类中心关系很大，而且对噪声和异常值敏感；而基本的 LDA 主题分析模型只能获得一个文档从属各个主题的概率，而不是类别标签的连续值，因此当一个文档从属多个主题的概率相近时，直接取最大概率对应的主题作为分类结果会不精确。

为此，我们提出利用将这两个模型结合的方法，能够扬长避短：LDA 主题分析

模型能够将数据量较大的词向量转换为简单的类别从属概率，不仅能够帮助 K-means 滤去噪声，而且降低了 K-means 的计算时间复杂度；K-means 能够对 LDA 模型的输出进行聚类，使得分类结果更加准确。

5.5 词嵌入向量模型

词向量是指将文本词语转化为带有语义信息的向量，使得计算机可以读懂并对词语信息进行计算。早期词表示方法主要通过建立人工规则把词映射到高维向量空间中。例如，独热表示方法将字符串转化成整型向量，这种方法不能捕获词语之间的联系和区别，而且也不能利用文本上下文信息。此外，独热表示的每个词向量维度都是词典中的词数，因此其编码结果属于高维稀疏向量，严重增加了计算量。

本实验采用的是词嵌入向量模型 Word2Vec。该模型在 2013 年由 Google 发布，它是基于神经网络的训练方法，可以将词转换为词向量并计算各个向量之间的联系。Word2Vec 模型的输入层和隐藏层的权值矩阵经过训练能够得到代表某一词语的浮点型词向量，且维度远远小于独热编码等传统词向量模型。该模型提供了两种训练模型，分别为 CBOW 和 Skip-gram。下面简单介绍本实验采用的 CBOW。

CBOW 模型是指利用目标词的上下文词作为模型的输入来预测当前目标词，该模型结构如下图所示。

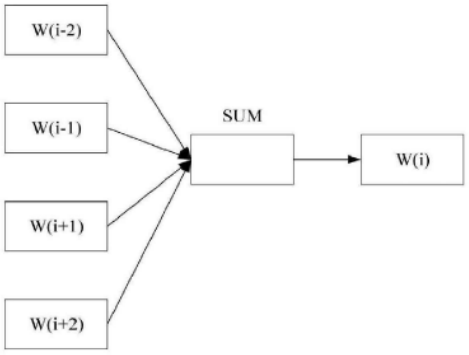

图 5.5.1—CBOW 模型示意图

CBOW 模型输入部分为目标词 w_i 的 n 个上下文词语。在文本的词序列中，模型选取文本序列中某一词 w_i 作为预测目标词，并将与预测目标词连续的上下文词语组成特征信息。假设模型只选择上文一词 w_{i-1}、本词和下文一词 w_{i+1} 作为模型输入，则模型公式如下：

$$h = E(w_{i-1} + w_i + w_{i+1})$$

$$y = \text{softmax}(h_k) = \frac{\exp(w_k{}' h_k)}{\sum_{k-1}^{K} \exp(w_k{}' h_k)}$$

其中 p_t 表示词语第 i 个类别出现的概率，E 代表权值矩阵，w_k 代表权值向量。

6 实验流程和结果分析

6.1 数据集介绍和处理

本实验采用的细粒度情感分析数据集是 AI-Challenger 2018 中文细粒度情感分析数据，该数据包含评论文本和该样本在服务、地理位置、店铺环境、菜品品质、总体体验等多方面的情感极性标签，并且已经按 8:1:1 的比例分为训练集、验证集和测试集。而对于菜品名称口味聚类，我们利用网络爬虫爬取部分商家菜品名称共 8000 条，并自行进行数据标注。

对于 AI-Challenger 2018 中文细粒度情感分析数据，它有以下特征：

- 文本内容杂乱，包含大量标点符号、格式符、表情符号 emoji 等无用信息
- 样本类别分布极其不均匀，以服务速度为例，“未提及”类别样本数为 86409，“正面”类别样本数为 12039，“中性”类别样本数为 1297，“负面”类别样本数为 4355，占比最大类的个数甚至是占比最小类的 66 倍
- 样本领域性强，由于这些数据来源于美团美食评论，包含大量菜品名称的专有名词，以及一些用于表示情感的网络用语，这些都是普通分词工具所不能有效处理的

而我们自己爬取到的菜品名称数据则相对比较好处理，一方面其数据量少，且每条数据通常只是一个简单的短语；另一方面，菜品名称结构简单，一般不包含无用的标点符号、格式符和表情符号 emoji。对于这 8000 条数据，我们采取 4 折交叉验证，并将验证性能的平均值作为模型最终的性能。

6.1.1 数据预处理

这两部分数据的预处理流程相同，具体为：

- 调用 Jieba 库，将各个文档进行分词
- 加载哈尔滨工业大学停用词语料，并进行一定程度的扩充，然后从分词结果中去除停用词、标点符号、格式符合表情符号 emoji
- 基于得到的分词结果构建词典，用于记录词到词编号的映射，从而将每个句子的字符序列转化为词编号的整型序列
- 将得到的整型序列输入到预训练的词嵌入向量模型中，得到输出的词嵌入向量，从而将句子的整型序列转化为包含语义信息的浮点数序列。

6.1.2 任务评价指标

本实验设计的系统的两个模块完成的都是分类任务，因此为了体现模型效果，我们采用了文本分类任务中的常用评价指标，其中包括准确率(Accuracy)、F1 值(F1-value)，以及各个网络在测试集上的计算时间。

准确率针对分类结果中获得正确分类的文本数量在总文本中的占比；F1 值为综合评价指标，由准确率和召回率(Recall)得到，它们的计算公式如下：

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1\_value = \frac{2Acc * \text{Recall}}{Acc + \text{Recall}}$$

其中，TP、TN、FP、FN 分别表示正例被预测为正例、负例被预测为负例、负例被预测为正例、正例被预测为负例的样本个数。

对于计算时间指标，AI-challenger 2018 提供的测试集数据量为 15000，我们可以根据模型在测试集上的计算时间来估计实际应用时的分析耗时。

6.2 实验方案、参数设置和结果分析

6.2.1 数据分布不均衡的处理方案

首先，考虑到 AI-Challenger 2018 的样本类别分布极度不平衡，而且在实际应用时，差评的重要性通常比好评要高。基于这些因素，我们必须在训练前先进行样本均衡。对此有以下方案：

- 简单欠采样，将样本数较多的类别的样本按一定概率丢弃
- 简单重采样，将样本数较少的类别的样本按比例重复
- 基于 K-means 的样本均衡，先将某一类的样本进一步细化聚类，再在每个聚类结果中简单欠采样或简单重采样
- 在数据预处理时统计各类别占比，以它们的倒数作为损失函数的加权值

对于简单欠采样方案，它会严重牺牲模型识别占比较大类的能力，并且对识别占比较小类的能力提升没有帮助；对于简单重采样方案，可能存在无用样本重复的现象，不仅不会对模型识别占比较小类有太大帮助，还延长了模型训练时间；对于第三个方案，K-means 聚类的时间复杂度为 $O(N^2)$，这对于十万余条训练数据来说是不可容忍的，而且实际聚类结果轮廓系数非常小，聚类效果较差。基于上述事实与分析，我们选择第四个方案，该方案处理简单，而且经实验证明的确能够在一定程度上缓解数据不均衡的问题。

6.2.2 实验方案

对于细粒度情感分析模型的对比实验，将我们的模型与一下方法进行比较：

1）CNN：利用不同大小的卷积核获取文本 n-gram 特征信息从而进行文本分类的基于二维卷积核的 CNN 模型；

2）LSTM：经典的长短时记忆网络，通过对隐藏层输出特征矩阵进行平均池化计算后，输出文本分类层进行分类；

3）BERT：预训练模型，一方面通过 masked language task 提升模型对语言序列特征的适应性，另一方面通过两组反向连接的 transformer 学习语言序列的正逆向序列特征。

4）TD-BiLSTM：由两个 BiLSTM 构成的网络结构，主要针对目标词上下文信息进行建模，并生成基于目标依赖的关系进行分类。

由于我们提出的菜品名称口味分类模型是开创性的，目前没有其他模型进行性能对比，我们选择直接将 LDA、K-means 和我们的模型进行性能对比。

对于细粒度情感分析模型的消融实验，我们通过分别屏蔽 text-CNN 各个并行的卷积通道、LSTM+多头自注意力机制、多头自注意力机制来检验各个模块对于模型性能的影响。

6.2.3 实验参数设置

表 1：细粒度情感分析模型参数设置

参数	参数值
batch_size	10
text-CNN 卷积核数	2,3,4,5
LSTM 的 hidden_size	256
LSTM 的 num_layer	1
学习率	0.0002
损失函数	加权交叉熵损失函数
迭代次数	60
词嵌入向量维度	300
优化器	Adam
训练设备	GPU

6.2.4 对比实验结果分析

表 2：不同细粒度情感分析模型对比结果

模型	准确率	F1 值	测试集计算时间
CNN	0.6231	0.5988	35.28 秒
LSTM	0.6642	0.6803	47.61 秒
BERT	0.7212	0.7147	302.79 秒
TD-BiLSTM	0.7349	0.7256	74.80 秒
本实验模型	0.7839	0.7795	52.37 秒

如上表所示，本实验提出的 text-CNN+LSTM+多头自注意力机制模型在测试集上的准确率、F1 值都是最高的，而且还保持了较快的分析速度，体现了该模型的性能优越性。通过对比 CNN 模型与 LSTM 模型，LSTM 在准确率和 F1 值上都要比 CNN 高，这说明虽然仅仅卷积两个相邻词的局部信息是有一定效果的，但是 LSTM 能更好地处理时间序列特征的语义信息，而在 LSTM 和 TD-BiLSTM 模型的对比上，后者效果更好更加证明了 LSTM 在细粒度情感分析模型的有效性。BERT 虽然与 CNN 和 LSTM 相比更加有效，但是该模型的计算复杂性太高，不适用于我们的应用场景。而本实验模型能够综合 CNN 和 LSTM 各自的特征提取特点，而且没有额外附加其他复杂高的模块，因而能够在分类性能和计算性能上取得双赢。

表 3：不同菜品名称口味分类模型对比结果

模型	准确率	F1 值	训练周期
仅用 LDA	0.7768	0.7244	30.67 秒
仅用 K-means	0.6190	0.6348	294.38 秒
LDA+K-means	0.8165	0.7611	65.12 秒

如上表所示，本实验提出的 LDA+K-means 模型在样本集上的准确率、F1 值都是最高的，并且保持了良好的计算速率。其原因正如前文所说，LDA 和 K-means 能够相互补足，K-means 能够纠正 LDA 不能精确分类的样本，而且在 LDA 输出类别概率后进行 K-means 也不会带来太大的时间损耗，这是因为类别概率与词嵌入向量维度要低很多。

6.2.5 细粒度情感分析模型的消融实验结果分析

我们通过屏蔽细粒度情感分析模型的各个子模型，得到以下性能对比结果：

表 4：细粒度情感分析屏蔽各模块性能结果对比

屏蔽模块	准确率	F1 值	测试集计算时间
Text-CNN 中卷积核长为 2 部分(后续用 text-cnn2 代替)	0.6897	0.6709	48.95 秒
Text-cnn3	0.7313	0.7125	50.73 秒
Text-cnn4	0.7449	0.7347	51.69 秒
Text-cnn5	0.7501	0.7456	51.98 秒
LSTM+MHSA	0.6575	0.6322	37.06 秒
MHSA	0.7422	0.7380	51.48 秒
不屏蔽(原模型)	0.7839	0.7795	52.37 秒

我们从两个角度来分析实验结果：

- 从分类结果准确率和 F1 值来看，所有的模块都能对模型性能起到一定的提升作用，其中占主导地位的是 Text-cnn2 和 LSTM 模块。这是因为，一方面，大多数修饰词和被修饰词还是相邻的（即在长度为 2 的卷积核中就可以被提取出来）；另一方面，LSTM 能够提取 CNN 无法处理的文本序列特征，并且具有一定的文本全局特征的分析能力。其他 Text-CNN 模块能够对 Text-cnn2 起到补充作用，而 MHSA 能够提炼 LSTM 提取的序列特征，因此它们都能够对模型性能提升起到帮助。
- 从计算时间来看，模型计算速率的瓶颈应该是 LSTM 模型，运算时间较长是 LSTM 与 RNN 相比的一个比较显著的缺憾。

7 实验结语

7.1 总结

对于细粒度情感分析，本实验在现有研究方法的基础上，针对线上平台中文评论文本数据细粒度情感分析任务进行研究。本实验工作总结如下:

- 对中文文本细粒度情感分析的方法、应用以及不足进行分析，根据本实验研究内容，运用深度学习分析方法实现了对于文本中多个方向词及其对应情感倾向。
- 为解决文本多方面情感分析中方面信息与情感信息对应不准确、以往模型对文本序列各个特征信息提取不足的问题，提出了 text-CNN+LSTM+MHSA 模型。该模型利用 LSTM 学习连续序列的特点和区域 CNN 捕捉局部信息的特点，将它们结合并在模型中融入注意力机制对重要信息赋予权重值，能够突出特定词与文本情感信息的联系。该模型不需要额外附加外部方面信息，运用不同的参数即可对一条文本的不同方面进行情感分析。通过对比实验和消融实验的证实，本实验模型性能更加良好，计算速率也有一定保障。

对于菜品名称口味分类，本实验开创性地提出了基于 LDA 主题模型+K-means 聚类分析的方法，通过分析商家菜单中的菜品名称分析口味分布，得到商家口味向量，与特定用户的口味偏好向量计算相似度获取商家口味评分。通过对比实验的证实，本实验的模型性能优良，计算速率也有一定保障。

7.2 展望

随着线上平台规模不断发展，商家和消费者更加重视平台积累的海量文本内容背后所隐藏的价值，从海量的评论文本中如何获取更加准确的分类模型，从海量的菜品名称中如何获取更加准确的口味分析模型，具有重要的现实意义和研究意义。本实验系统所提出的两个子模型的性能尽管比较客可观，但仍有许多不足，未来进一步研究可以在以下几个方面进行:

- 本实验的细粒度情感分析研究是基于线上美食平台公开的数据集开展的，由于现在研究缺乏标注的中文数据集，在模型训练上具有局限性。未来下一步工作是要基于建立适合深度学习研究的其他领域的基准数据集，进一步优化模型的分析能力和泛化能力，提高领域自适应性。
- 本实验的菜品名称口味分析的数据集是自行爬取、自行标注的，数据量偏少，主观性较强，而且市面上缺乏店家自主标注的菜品口味标注，且标注规格不统一。未来可以建立相关方面更完备的数据集，使得模型得到更完善的训练。

参考文献

[1] Yejin Tan, Guo Wangshu, He Jiawei, Liu Jian, Xian Ming. A Fine-grained Sentiment Analysis Based on Dependency Tree and Graph Attention Network[J]. Journal of Physics:Conference Series, 2020, 1651(1).

[2]黄胜，Web 评论文本的细粒度意见挖掘技术研究[D].北京：北京理工大学,2018.

[3]王海燕,陶皖,余玲艳,王鸣鹃.文本细粒度情感分析综述[J].河南科技学院学报(自然科学版),2021,49(04):67-76.

[4]Lin Gan. Research on Prediction of Different Categories of Video based on YOUTUBE Using Text Mining and Sentiment Analysis[C]//.Proceedings of 5th International Conference on Economics and Management, Education, Humanities and Social Sciences (EMEHSS 2021).

[5] Zhifang Fan, Zhen Wu, Xin-Yu Dai, Shujian Huang, Jiajun Chen. Target-oriented Opinion Words Extraction with Target-fused NeuralSequence Labeling[J]. Proceedings of the 2019 Conference of the North,2019.