面向多模态情感分析的统一框架探索：论文复现及优化

卢星宇　2211287　1362064411@qq.com[1]
赵若轩　2212030　172803403@qq.com[2]
李雅帆　2213041　2727035698@qq.com [3]
[123] Nankai University

摘要

情感分析（$SentimentAnalysis, SA$），尤其是基于目标的情感分析（$TBSA$）和多模态情感分析，近年来在自然语言处理（$NaturalLanguageProcessing, NLP$）领域得到了广泛关注。传统的 $TBSA$ 任务通常分为意见目标提取和目标情感分类两个子任务，这两者在处理时常采用管道化方法。然而，这种方法存在较大的局限性，尤其是任务间的错误传播和信息利用不足，导致模型性能难以提升。因此，探索更集成化的解决方案成为当前研究的重点。已有研究表明，当子任务具有强耦合关系时，集成模型能够显著提高任务的表现，尤其是在目标情感分析任务中，意见目标的提取与情感分类之间存在密切联系。另一方面，随着多模态数据的广泛应用，情感分析逐渐从单一的文本分析拓展到包括视觉、声音等模态的多模态情感分析。现有的多模态情感分析方法大多将情感分数预测作为一般的机器学习任务进行处理，忽视了情感极性（$Polarity$）和强度（$Intensity$）这两个关键方面。然而，情感分数本质上可以被分解为极性和强度，且这两者在情感分析中具有重要的互补作用。为了应对这些挑战，本文通过复现和优化现有的多模态情感分析框架，提出了一种集成化的端到端模型，旨在有效融合不同模态的信息并处理目标情感分析中的任务耦合问题。特别地，我们采用基于文本的共享-私有框架（$TCSP$）进行多模态情感分析，强调文本在情感分析中的主导作用，并通过共享语义和私有语义的结合提升模型的鲁棒性和准确性。通过这种方式，我们探索了多任务学习在情感极性与强度分类中的潜力，期望为多模态情感分析任务提供一种更为高效的解决方案。

关键词：情感分析；多模态；$TBSA$；情感极性；情感强度；$TCSP$

1. 引言 *Introduction*

情感分析作为自然语言处理领域的重要任务，近年来在多个实际应用中得到了广泛的关注，特别是在产品评论分析、社交媒体监测等领域。情感分析可以帮助企业了解消费者对产品或话题的看法，从而为决策提供支持。传统的情感分析任务大多侧重于文本数据，然而，随着多模态信息的广泛使用，情感分析逐渐拓展到了包括视觉、声音和文本等多种模态的信息融合。多模态情感分析不仅能够提高情感识别的准确性，还能更全面地理解人类情感的复杂性。

基于目标的情感分析（$TargetedSentimentAnalysis, TBSA$）作为一种更加细化的情感分析方法，致力于识别文本中针对特定目标（如产品或服务）的情感倾向。$TBSA$ 任务通常分为两个子任务：意见目标提取（$OpinionTargetExtraction, OTE$）和目标情感分类（$TargetSentimentClassification, TSC$）。传统上，这两个子任务是分别独立解决的，采用管道化方法将意见目标提取与情感分类分开处理。然而，这种管道化方法存在诸多问题，例如，前一个子任

务的错误可能会传递到后一个子任务，导致情感分类结果不准确。此外，管道化方法未能充分利用任务之间的内在联系，导致信息利用不充分。因此，近年来，越来越多的研究开始探索更为集成的解决方案，以端到端的方式处理 $TBSA$ 任务。

另一方面，随着社交网络和在线互动的日益普及，多模态情感分析（$MultimodalSentimentAnalysis, MSA$）逐渐成为情感分析研究中的一个热点。传统的多模态情感分析大多将情感分数预测作为一个通用的机器学习任务，而忽视了情感的极性（$Polarity$）和强度（$Intensity$）这两个维度的重要性。情感极性判断可以帮助判断情感的方向（正面、负面或中立），而情感强度则反映了情感的强烈程度。理解这两个维度对提高情感分析的性能至关重要。为了进一步提高情感分析的效果，最近的研究开始探索多任务学习框架，并在该框架中同时考虑情感极性、强度和情感分数预测任务。

本文旨在通过复现并优化现有的情感分析方法，探索更集成化的解决方案，旨在通过端到端的模型处理 $TBSA$ 任务，并在多模态情感分析中引入极性和强度的分类。我们提出了一种基于文本的共享-私有框架（$Text-CenteredShared-PrivateFramework, TCSP$），在该框架中，文本模态被视为主导模态，而视觉和声音模态则提供共享语义和私有语义，以补充文本模态并增强情感分析的效果。通过这种集成框架，我们旨在克服现有研究的局限性，提供更有效的情感分析解决方案。

2. 相关工作 *Relatedwork*

- 基于目标的情感分析（$TBSA$）[1]是情感分析中的一个重要研究方向，其任务包括意见目标提取（OTE）和目标情感分类（TSC）。在早期的研究中，$TBSA$ 任务通常采用管道化的方式，先进行意见目标的提取，再进行情感分类 [13]。这种方法虽然在一定程度上能够实现情感分析，但由于任务之间的信息传递问题，往往会导致最终结果的准确性受到影响。随着研究的深入，越来越多的工作开始探索集成方法，以避免管道方法中的缺陷。例如，Xu 等人提出了一种基于深度学习的联合模型，试图同时处理意见目标提取和情感分类任务 [19]。此外，Li 等人 [9]提出了一种利用上下文信息的模型，旨在改善目标情感分类的效果。这些方法虽然有效提高了任务的性能，但仍未能够彻底解决任务之间的耦合问题。Mitchell 等人 [10]提出的统一标记方案是另一种尝试，旨在通过统一标记方式将意见目标提取与情感分类整合在一起，但其效果仍未超越传统的管道方法。而集成化方法，尤其是在任务间具有强耦合关系的情感分析任务中，显示出了其较管道化方法的优势。近年来，许多研究集中在如何通过集成化方法提高情感分析的性能。
- 多模态情感分析[2]是近年来情感分析研究的一个重要分支，主要致力于融合文本、视觉和声音等多种模态的信息，以提升情感分析的准确性。Zadeh 等人 [2]提出了一种多模态情感分析方法，通过融合不同模态的特征来预测情感分数。这种方法强调了多模态特征的交互作用，但在情感极性和强度的具体分类上仍有所忽视。与传统的情感分析方法不同，最近的研究开始关注情感的极性（Polarity）和强度（Intensity），并提出了多任务学习框架，试图同时解决多个任务，如情感极性判断、强度预测以及情感分数预测 [7]。这种方法的优势在于，通过多任务学习能够提升模型对情感的细致理解，并促进情感极性与强度的有效预测。然而，现有的多模态情感分析研究通常将所有模态视为平等的，未能考虑不同模态在情感分析中的重要性差异。
- 基于文本的共享-私有框架（$TCSP$）[3]，该框

[1]李雅帆：A Unified Model for Opinion Target Extraction and Target Sentiment Prediction 论文复现以及优化 github 仓库链接：https://github.com/SerendipityFanan/Natural-Language-Processing

[2]赵若轩：Polarity and Intensity: the Two Aspects of Sentiment Analysis 论文复现以及优化 github 仓库链接：https://github.com/Pork-stuffing/NLP.git

[3]卢星宇：A Text-Centered Shared-Private Framework via Cross-Modal Prediction for Multimodal Sentiment Analysis 论文复现以及优化 github 仓库链接：https://github.com/xichenyao/nlp_final.git

架强调文本模态在情感分析中的主导作用，并通过共享语义和私有语义的融合来提升模型性能 [2]。共享语义有助于加强文本模态的语义表达，而私有语义则能够提供不同的视角来补充文本信息，从而提高情感分析的效果。

3. 问题定义 $Problemdefinition$

- 情感倾向性分析是情感分析中最常见的任务之一，目的是判断文本（如产品评论、社交媒体帖子等）表达的情感是正面的、负面的还是中立的。这通常被建模为一个分类问题，其中模型需要将文本分配到预定义的情感类别中。
- 情感强度分析除了判断情感的正负之外，还关注情感表达的强度或程度。这通常涉及到对情感的量化，例如使用连续的评分（如 1 到 5 星）或有序的类别（如弱、中、强）来表示情感的强度。
- 目标或方面级情感分析不仅要识别文本中的情感倾向和强度，还要确定这些情感是针对哪些特定目标或方面的。
- 多模态情感分析考虑了除了文本之外的其他信息源，如语音、面部表情、身体语言等，以更全面地理解和分析情感。这种分析通常涉及到跨模态信息的融合和交互，以捕捉更丰富的情感表达。
- 情感分析中的多任务学习可以通过同时学习相关任务来提高模型的泛化能力。例如，同时进行情感倾向性分析和情感强度分析，或者将情感分析与其他 NLP 任务（如方面提取、意见目标提取）结合起来。

4. 方法 $Methodology$

4.1. A Unified Model for Opinion Target Extraction and Target Sentiment Prediction [8]

4.1.1 任务定义与目标

将完整的基于目标的情感分析（TBSA）任务定义为序列标注问题，采用统一标签集 $Y^S = \{B-POS, I-POS, E-POS, S-POS, B-NEG, I-NEG, E-NEG, S-NEG, B-NEU, I-NEU, E-NEU, S-NEU\} \cup \{O\}$。其中，除 O 外，每个标签包含目标提及边界和目标情感两部分信息，例如 $B-POS$ 表示正向目标提及的开始，$S-NEG$ 表示单字负向意见目标。

对于给定长度为 T 的输入序列 $X = \{x_1, \cdots, x_T\}$，目标是预测标签序列 $Y^S = \{y_1^S, \cdots, y_T^S\}$，其中 $y_i^S \in Y^S$。

4.1.2 复现流程

本次实验旨在对 E2E-TBSA 模型进行完整的论文复现与实验流程再现。本复现过程涉及以下几个关键步骤：

1. **数据准备与预处理**
   process_data.py 脚本负责将原始 XML 标注数据（如 SemEval 数据集）解析为 (word, tag) 序列格式：
   - 脚本使用 scrapy.selector.Selector 对 XML 中的 <sentence> 和 <aspectterm>（或 <opinion>）标签进行解析提取。
   - 通过 extract_aspect 函数将句子中的观点目标替换为占位符（如 ASPECT0），并用 process_text 清洗文本，对标点、大小写、特殊字符进行处理。
   - 最终使用 word_tokenize 将句子切分为词，并根据目标词的极性(POS/NEG/NEU)或无目标（O）为每个词标注标签，将处理结果输出至./data/ 目录下的.txt 文件。
2. **数据加载与标签处理**在 utils.py 中定义了一系列数据处理函数，包括从分割后的文本中读取数据 (read_data)、构建词典 (get_vocab)、设置词与字符索引 (set_wid, set_cid)、根据指定的标注体系（OT、BIO、BIEOS）对标签进行格式转换（ot2bio, ot2bieos 等）以及加载情感词典（mpqa_full.txt）和预训练词向量 (load_embeddings)。通过这些函数，我们能将原始数据转化为模型可直接处理的格式（包括索引化、窗口化输入、字符级信息、情感感知语

言模型辅助标记等)。

3. **模型构建流程**在 model.py 文件中定义了 E2E-TBSA 模型的核心结构。模型采用 DyNet 框架实现，核心流程为：

   (a) 初始化模型参数与模块

   当 main.py 中完成数据集处理(build_dataset)和预训练词向量加载(load_embeddings)后，会实例化 Model 类。

   在 Model 的构造函数中，首先创建 ParameterCollection 以管理模型参数，然后根据超参数与词典信息初始化如下模块：

   - 嵌入层：使用 WDEmb 为词提供预训练词向量表示，可根据 use_char 决定是否使用 CharEmb 提取字符级特征。
   - 编码层：通过双向 LSTM 对序列进行上下文编码，其中 lstm_ote 层针对 OTE 任务提取特征，lstm_ts 则将 OTE 提取的结果进一步融合，用于 TS 预测。
   - 输出层与参数：使用 Linear 层 (fc_ote, fc_ts) 将 LSTM 隐状态映射到对应任务标签空间。模型还定义了 transition_scores 与 W_trans_ote，用于在 TS 预测中融入 OTE 的边界置信信息。

   根据参数 optimizer，模型会选择合适的优化算法（如 SGD 或 Adam）。此时，模型的参数与组件初始化完成，具备接受输入并生成预测结果的能力。

   (b) 前向计算与损失函数（Model.forward）

   模型训练时,main.py 中的 run 函数会调用 model.forward 来获取当前输入句子的损失和预测结果。

   在 forward 函数中主要步骤如下：

   - 计算图重构：调用 dy.renew_cg() 重建计算图。
   - 嵌入提取：根据 wids（与可选的 cids）获取词和字符嵌入，将它们拼接获得输入向量序列。
   - LSTM 编码：通过 lstm_ote 与 lstm_ts 对嵌入序列进行双向 LSTM 编码，获得针对 OTE 和 TS 的隐状态表示。
   - OTE 预测与损失：对 OTE 相关的隐状态通过 fc_ote 计算标签分布 p_y_x_ote，与 gold_ote_labels 对比求得 OTE 损失。
   - TS 预测融合 OTE 信息：通过 W_trans_ote 将 OTE 分布投射到 TS 标签空间，并与独立的 TS 分布 p_y_x_ts 加权融合得到 p_y_x_ts_tilde，进而计算 TS 损失。
   - 情感感知任务损失：额外的 fc_stm 输出与 gold_stm_labels 产生 STM 损失，用于辅助训练，提升情感上下文感知能力。

   最终总损失为 OTE、TS 和 STM 损失之和。调用 loss.backward() 和 model.optimizer.update() 即可完成参数更新。

4. **模型训练流程**

   (a) 数据集与初始化：在 main.py 中，首先根据用户参数指定数据集名称（-ds_name）和超参数（-n_epoch, -dropout 等），然后通过 build_dataset 得到 train/val/test 数据集，并加载预训练词向量生成 embeddings。

   (b) 数据准备与初始设置：在调用 build_dataset 和 load_embeddings 后，已有处理好的训练、验证和测试集，以及加载完成的词嵌入。

   (c) 训练循环：每个 epoch 开始对训练集打乱，逐条输入至 model.forward 计算损失并更新参数。训练中可对学习率执行简单衰减，避免过拟合。

(d) 验证与保存模型：每个 epoch 结束后在验证集上评估若性能提升，则存储当前最佳模型参数 (model.pc.save)。训练完成后对测试集评估并输出预测结果用于后续分析。

5. **复现难点与细节**

(a) 环境配置与版本一致性: 该模型在实现和调试过程中使用特定版本的 Python、DyNet（如 DyNet 2.0.2）、以及特定的 CPU 环境和随机种子（dynet_seed 与 random_seed）以保证结果可重复性。为成功复现实验结果，需严格对照作者设定的运行环境与版本信息，确保:

- 使用与原作者相同或兼容的 Python 版本与依赖包版本。
- 在 CPU 环境下运行并设置与作者一致的随机种子以减少结果波动。
- 下载和使用正确路径下的预训练词向量文件、数据集与情感词典，保证输入数据与代码期望完全匹配。

(b) 数据与标签格式转换: 原始数据的目标及情感标注格式与模型所采用的 BIEOS 或 BIO 标注方案存在差异。复现时需要通过 utils.py 中的函数（如 ot2bieos、bio2ot）将 OT 格式转换为适合序列标注的标准化标签序列。初次接触这些函数时可能不知其逻辑与意图，需要仔细阅读代码和注释才能确保标签格式与原论文一致。

(c) 情感词典与辅助特征整合: 模型加入 mpqa_full.txt 情感词典辅助特征（如 stm_lm_labels）的建模。这类情感感知语言建模特征在 utils.py 的 set_lm_labels 函数中实现，其逻辑相对复杂。若对其原理不清楚，则难以确保与原作者相同的特征输入方式。在复现时需明确理解这些特征的计算流程及在 model.py 中的使用方式。

4.1.3 创新方法

我在现有的 E2E-TBSA 模型基础上，引入了更强大的预训练语言模型（PLM）作为词表示层，以取代原有的静态词向量（如 GloVe）。具体创新点如下:

1. 引入上下文感知的词嵌入：传统词向量（例如 GloVe）为每个词提供固定的嵌入，与上下文无关。而通过使用 BERT 等预训练语言模型，本方法可为每个词生成基于上下文的动态嵌入，能更好地捕捉评论句子中的细微语义差异。
2. 深度整合预训练模型输出与现有结构：我将 BERT 输出的隐藏表示直接作为模型的输入特征，将其与 BiLSTM 以及 CRF 等组件有机结合。在这种整合下，BERT 负责为每个单词生成高质量的上下文相关表示，而 BiLSTM 与 CRF 则在此基础上对序列信息和标签转移结构进行建模，从而提升目标抽取与情感分类的性能。
3. 可选的参数微调策略：除直接使用 BERT 的固定表示外，本方法还可通过在训练过程中对 BERT 参数进行微调，使其更适应特定的目标/情感分析任务场景。通过调优预训练模型参数，有望进一步提高整体模型的性能和稳定性。

综上，本研究的创新点在于利用预训练语言模型实现上下文感知的词表示替换传统静态词嵌入，并在此基础上保留或融合现有的深度序列标注结构（如 BiLSTM + CRF）。这种方法有助于在目标级情感分析任务中获得更好的性能和更强的领域适应性。

4.2. Polarity and Intensity: the Two Aspects of Sentiment Analysis[14]

4.2.1 实验准备

在https://github.com/Jie-Xie/CMU-MultimodalDataSDK.git上下载 mmdata 包并导入，在这个包中包含对 MOSI 数据集的下载、分类、以及调用。

4.2.2 复现方法

论文提出构建单模态和多模态的单/多任务学习模型，将情感分数预测作为主要任务，同时将情感的极性和/或强度分类作为辅助任务。对论文的复现我们从单模态和多模态来分别进行。

1. *unimodal*

   单模态模型的构建主要分为三个方面，语言模态（Verbal），声音模态（Vocal），视觉模态（Visual）。不同模态除了模型的输入需要进行修改，实现方法其实大体相同，在这里我们从不同的任务角度来实现复现代码。

4.2.3 单任务情感回归（情感）

定义评价指标. 在函数 pearson_cc 中自定义 Pearson 相关系数（相关性衡量指标）。该指标用来评估模型预测值与真实值之间的线性关系，值越大表示预测越准确。

数据预处理. 定义 pad 函数用来填充或截断序列，确保在输入模型的数据序列长度一致。调用 mmdata 中的 MOSI 接口来加载 MOSI 数据集的多个模态特征（音频特征 covarep、面部表情特征 facet、词嵌入 embeddings）以及情感标签 sentiments（表示情感的 Valence 值）。接着用 train_ids、valid_ids 和 test_ids 分别包含了训练集、验证集和测试集的 ID 列表。通过合并词嵌入（embeddings）和音频特征（covarep），创建了一个双模态数据集，并使用 align 方法对齐数据，确保每个视频段（segment）对应相同的特征维度。遍历 train_ids、valid_ids 和 test_ids，筛选出同时包含音频特征（covarep）和词嵌入特征（embeddings）的有效数据，构建训练集、验证集和测试集。

使用 pad 函数对训练、验证和测试数据进行填充，确保每个序列的长度一致。提取情感标签（sentiments）作为目标变量 y_train、y_valid 和 y_test。对音频特征（covarep）进行归一化处理，将其值缩放到 [0, 1] 范围内。

构建模型. 模型的输入是一个形状为 (maxlen, 74) 的张量，表示每个样本的音频特征序列。网络层包含多个全连接层（Dense）和 Dropout 层来防止过拟合。最后的输出层是一个回归层，使用 tanh 激活函数来预测情感值（Valence）。使用 Adamax 优化器进行模型训练，损失函数选择均方误差（mae）。评估指标为皮尔逊相关系数（pearson_cc）和均方误差（mae）。

训练模型. 训练模型时设置批次大小和训练轮次。使用早停（EarlyStopping）策略来防止过拟合。

评估模型并输出结果. 在训练集、验证集和测试集上评估模型性能，输出皮尔逊相关系数和均方误差。使用训练好的模型进行预测，并将测试集的预测结果保存到文件中。

4.2.4 多任务情感回归（情感 + 强度）

在构建模型之前的准备，如评估指标、数据预处理都与单任务的情感回归没有大的差异，这里我们对情感标签以及强度标签的处理做出分析。

情感标签（Valence）处理. 情感值是一个连续的数值，通常范围在 -5 到 +5 之间，表示从非常负面到非常正面的情感。将这些情感标签存储在 y_train 中。

强度标签（Intensity）处理. 在论文的分析中，首先根据情感值的绝对值将每个样本的情感强度分类为四个等级：

- 强烈情感（strong）：情感值的绝对值大于等于 2.5，强烈情感被表示为 [0, 0, 0, 1]。
- 中等情感（medium）：情感值的绝对值大于等于 1.5，表示为 [0, 0, 1, 0]。
- 弱情感（weak）：情感值的绝对值大于等于 0.5，表示为 [0, 1, 0, 0]。
- 中性情感（neutral）：情感值的绝对值小于 0.5，表示为 [1, 0, 0, 0]。

将每个视频片段的情感强度标签添加到 z_train 列表中。每个标签是一个 one-hot 编码，长度为 4，其中每个位置表示不同的情感强度类别。z_train 最终是一个列表，包含了每个训练样本

对应的情感强度标签（one-hot 编码形式）。

构建模型.

- Input 层：输入的形状是 (maxlen, 74)，即每个样本由 74 维的特征组成，序列的长度是 maxlen（通常为 15）。这些特征是音频数据的处理结果。
- Dropout 层：为了防止过拟合，在模型的输入层后加入了一个 Dropout 层，设置丢弃率为 0.2，意味着每个训练步骤中 20% 的神经元将被随机忽略。
- Dense 层：这里有三个全连接层，每个层包含 32 个神经元，并使用 ReLU 激活函数。每一层的输出作为下一层的输入，通过这种方式提取特征。
- Flatten 层：该层将多维的输入（如二维矩阵）展平为一维向量，以便连接到后续的全连接层。
- main_output 层：模型的主要输出层，目标是回归任务，输出一个 tanh 激活的单一实值，用于预测情感值（Valence）。tanh 激活函数通常用于回归任务，可以输出 [-1, 1] 区间的值。
- auxiliary_output 层：这是辅助任务的输出，用于分类任务。该层的输出使用 softmax 激活函数，用于输出情感强度的四个类别（strong, medium, weak, neutral）。softmax 激活将输出值转化为概率分布，因此这里有 4 个神经元分别对应情感强度的四个类别。

编译模型. 这里采用了一个简单的多层感知机（MLP）结构，包括了 Dropout 和 Dense 层。主输出（Valence）采用 tanh 激活函数进行回归，输出连续值。辅助输出（Intensity）采用 softmax 激活函数进行四分类任务。使用 Adamax 优化器和均方误差（MAE）作为回归任务的损失函数，分类任务使用交叉熵损失函数。

训练模型. 模型训练采用多任务学习（Multitask Learning）策略，同时优化情感的回归任务和情感强度的分类任务。同样使用 EarlyStopping 来防止过拟合，若验证损失连续 5 轮没有改善，则停止训练。

评估方法与单任务模型相同。

4.2.5 多任务情感回归（情感 + 极性）

对于情感 + 极性的多任务情感回归，情感分析同样是一个回归任务，用来预测情感值，与情感 + 强度模型一致。但是对于极性来说，这是一个二分类任务，极性标签是根据情感值是否为正来进行二值化，情感值大于等于 0 视为“正极性”，设置标签为 1，小于 0 视为“负极性”，设置标签为 0。

数据预处理. 与前两个模型不同的地方在于，这个模型将音频特征（covarep）和文本嵌入（embeddings）被拼接成 x_train、x_valid 和 x_test，并对其进行填充或截断处理。并对音频特征进行归一化，使其值的范围在 [0, 1] 之间。

构建模型. 使用了 Dense 层和 Dropout 层来构建一个简单的前馈神经网络。Dropout 用于防止过拟合，Dense 层用于学习特征的非线性组合。这个模型使用了多个 Dense 层和 Flatten 层，网络结构相对较简单，适用于处理较少的特征或非时序数据。

编译模型. 使用 Adamax 优化器，调整学习率、动量等超参数. 对情感回归任务使用 mae（均方误差），对极性分类任务使用 binary_crossentropy（二分类交叉熵）。

模型输出. 主输出（main_output）：预测情感（回归任务），使用 tanh 激活函数，输出一个连续的情感值。辅助输出（aux_output）：预测情感极性（分类任务），使用 sigmoid 激活函数，输出一个 0 或 1 的标签，表示正极性或负极性。

4.2.6 多任务情感回归（情感 + 强度 + 极性）

对于三任务的情感回归来说，这个模型基本上就是将前面提到的多个任务（情感回归、极性分类、强度分类）合并起来实现的。它是通过一个

多任务学习（MTL，Multi-Task Learning）框架来同时处理多个任务，使用共享的网络层来提取特征，再为每个任务设置独立的输出层。

输入层和共享网络层： all_input 是输入层，接收形状为 (maxlen, 74) 的数据（每个序列包含 74 个特征，每个序列有 maxlen 个时间步）。这些输入数据包含了音频、面部表情等信息。然后通过几个 Dense 层（h2, h3, h4）和 Dropout 层进行特征提取。

主任务输出（情感回归任务）： main_output 层用于情感回归任务，即预测情感的强度（Valence）。它是一个回归任务，输出一个连续值，使用 tanh 激活函数。

辅助任务 1 输出（极性分类任务）： auxiliary_output_1 层用于极性分类任务，即预测情感的极性（positive/negative）。它是一个二分类任务，输出一个概率值，使用 sigmoid 激活函数。

辅助任务 2 输出（强度分类任务）： auxiliary_output_2 层用于情感强度分类任务，即根据情感强度的大小进行分类（如强、中、弱、无情感）。它是一个多分类任务，输出一个 4 类的概率分布，使用 softmax 激活函数。

损失函数和损失权重： 主任务（情感回归）的损失使用 mae（均方误差），而两个辅助任务使用适当的损失函数：极性分类使用 binary_crossentropy，强度分类使用 categorical_crossentropy。为了平衡三个任务的训练，模型通过 loss_weights 来调整各任务的权重。

到此我们已经实现了单模态的多任务情感回归，接下来对多模态进行复现。

2. *multimodal*

在实现多模态的情感分析时，总体来说和单模态的实现方法相同，根据单/多任务分别进行了不同的模型实现。但是由于是多模态，需要考虑不同模态的特征的融合策略。论文中提到了四种融合策略，分别为早期融合（EF）、晚期融合（LF）、张量融合网络（TFN）、层次融合（HF）。单任务和多任务的情感分析这里不作区别，我们主要分析不同融合策略改如何构建模型。

早期融合（EF）： 在输入阶段就将不同模态（如视觉、音频、文本）的特征进行拼接（concatenate），然后一起传入模型进行训练。即将不同模态的特征在输入层进行融合，早期整合信息后由一个统一的网络进行处理。

晚期融合（LF）： 模型的输入是三个模态的数据，分别是 covarep_layer_0（音频）、facet_layer_0（视觉）和 text_layer_0（文本）。每个模态都经过一系列的处理（如 Dropout、Dense 层等）后再进行融合。每个模态的学习是独立的，最后融合后进行最终预测。

张量融合网络（TFN）： TFN 采用了张量点积（Tensor Dot Product）的方式来融合多模态的特征。具体来说，它结合了音频、视觉和文本特征来进行情感分析。

- 音频和视觉模态的融合：通过点积融合层 dot_layer1，将音频特征（covarep_layer_6）和视觉特征（facet_layer_6）进行张量点积融合。
- 三模态融合：然后，将音频和视觉模态融合后的特征（dot_layer1_reshape）与文本特征（text_layer_4）再次进行点积融合。这里使用了 dot_layer2 进行点积操作。
- Reshape 操作：在点积后，使用 Reshape 层来调整数据的形状，以适应后续的 LSTM 层：
- LSTM 和后续的 Dense 层：融合后的特征传递给 LSTM 层（LSTM(128)）进行进一步处理，最后通过多个全连接层（Dense）输出最终的情感预测结果（main_output）和强度分类结果（aux_output）：

层次融合（HF） 模型的核心思想就是通过层次化的方式将三种模态的信息进行逐步融合。在此过程中，先通过 LSTM 层进行时序建模，再通过全连接层进一步提取特征，并通过 Dropout 来防止过拟合。具体来说，模型首先在较低层次融合声音和视觉特征（covarep_layer_5 和

facet_layer_6)，然后在更高层次进一步融合语言特征（text_layer_0）。这种层次化的融合方式允许模型在不同的层次上学习不同类型的信息。因为先前的研究 [22]表明语言模态在单模态情感分析中最为有效，而声音模态效果最差。

3. **复现难点与细节**

这篇论文的复现难点其实在于数据的获取以及处理，这篇论文发表在 2018 年 ACL 挑战赛上，当时 CMU 将 MOSI 数据集分为不同的特征文件，公布在了学校网站上，一开始为了处理数据做了不少努力，但是后面发现在 GitHub 上可以找到开源的处理数据的包，一切问题也就迎刃而解。

在具体实现模型代码时工作量较大，因为这篇论文对不仅是单/多模态，并且还有单/多任务的区分，所以在具体实现时要做好规划。我首先复现了单模态的单任务模型，其次对多任务模型进行复现。复现过程中模型的参数以及特征的选择都是非常重要的。

在多模态的复现时，有了单模态的经验，其实最重要的是如何对融合策略进行实现。在实现 TFN 以及 HL 融合策略时，因为原理较为难懂，所以在实现这一部分时花了较长的时间。

4.3. A Text-Centered Shared-Private Framework via Cross-Modal Prediction for Multimodal Sentiment Analysis[18]

4.3.1 复现方法描述

1. **跨模态预测**

任务定义：

给定一个表示为 $\mathrm{x}_l = \{x_l^t : 1 \leq t \leq L, x_l^t \in \mathbb{R}^{d_l}\}$ 的文本特征序列，目标是预测对应的视觉或声学特征序列，表示为 $\mathrm{x}_i = \{x_i^t : 1 \leq t \leq L, x_i^t \in \mathbb{R}^{d_i}\}, i \in \{v, a\}$。

预测模型：

- 模型框架: 具有注意力的 Seq2Seq 模型 [3]
- 编码器: 以文本特征 x_l 作为输入，
- 输出隐藏状态:$\mathrm{h}_{enc} = \{x_{enc}^t : 1 \leq t \leq L, x_{enc}^t \in \mathbb{R}^{d_h}\}$

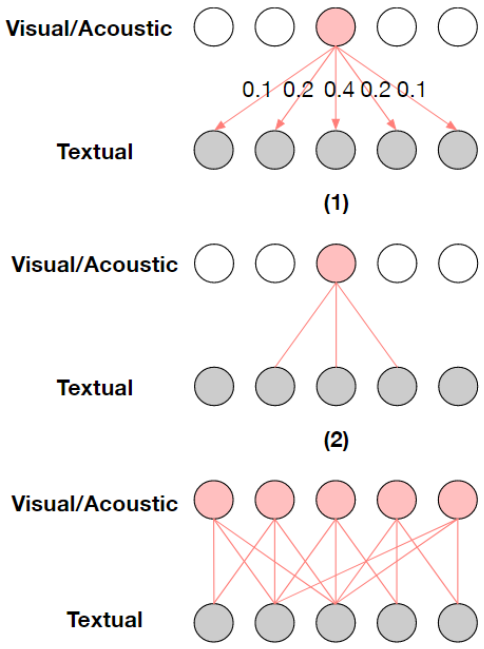

图 1. 从跨模态预测模型中获取共享掩码

- 解码器将编码器的先前隐藏状态 h_{enc}^{t-1} 和隐藏状态作为输入，并在间隔 t 预测非文本特征 $\mathrm{x}_i^t, i \in \{v, a\}$
- 损失函数:MSE
- 预测损失值: $\mathrm{e}_{l \to i} = \{e_{l \to i}^t : 1 \leq t \leq L\}$
- 预测模型的注意力图： $\mathrm{m}_{i \to l}$
- 编码器和解码器:LSTM[6]

共享掩码生成方式：

1. 注意力权重排序：对于每一行 t，对模型的注意力权重 $m_{i \to l}^{t,*}$ 进行排序，其中 i 代表视觉或声音模态。

2. 选择高权重索引：接着，我们筛选出前 K_s 个最大注意力权重的索引，记为 S^t。

3. 生成共享掩码：利用这些索引，构造共享掩码 $smask$，其维度为 $\mathbb{R}^{L*L}$。如果 t_1 存在于 S^{t_2} 中，则 $smask^{t_1,t_2}$ 被标记为 1，表示这两个特征之间存在共享信息；否则，标记为 0。

为了更清晰地阐释这一过程，通过图1进行了可视化展示。

私有掩码：

用于找出两种非文本模态的私有语义，捕捉每种非文本模态独有的特征。

私有掩码的创建过程：

1. 模态特征表示：首先，我们定义了一个话语的多模态特征，包括文本 (l)、视觉 (v) 和声学 (a) 模态，表示为 $x_i = \{x_i^t : 1 \leq t \leq L, x_i^t \in \mathbb{R}^{d_i}\}$，其中 L 是序列的长度，d_i 是模态 i 的特征维度。

2. 损失值计算：利用训练好的跨模态预测模型，我们计算从文本到视觉（$e_{l\to v}$）和文本到声学（$e_{l\to a}$）的预测损失。

3. 索引选择：对这些损失值进行排序，选择损失值最大的 K_p 个索引，这些索引对应的特征被认为是难以由文本模态预测的，即模态私有特征。

4. 私有掩码生成：基于这些索引，我们构建私有掩码 $pmask \in \mathbb{R}^L$，如果时间步 t 属于索引集 P，则 $pmask^t = 1$，否则为 0。

输入层：

使用三个 $LSTM$ 网络对输入的多模态特征 x_i 进行编码，产生 $h_i = \{h_i^t : 1 \leq t \leq L, h_i^t \in \mathbb{R}^{d_h}\}$。

$$\begin{aligned}
\mathrm{h}_l &= \mathrm{LSTM}_l(\mathrm{x}_l) \\
\mathrm{h}_v &= \mathrm{LSTM}_v(\mathrm{x}_v) \\
\mathrm{h}_a &= \mathrm{LSTM}_a(\mathrm{x}_a)
\end{aligned}$$

共享模块:

为了利用来自非文本模态特征的共享信息来增强单词的表示，提出使用掩码跨模态注意力网络：

计算每个单词的非文本表示 h_i, $i \in \{v, a\}$ 的注意力分数，将分数表示为 $s_{l\to i}$，评分函数的参数:$W_1, W_3 \in \mathbb{R}^{d_h \times 2d_h}$, $W_2, W_4 \in \mathbb{R}^{1\times d_h}$, $b_1, b_3 \in \mathbb{R}^{d_h}$

$$\begin{aligned}
\mathrm{s}_{l\to v}^{t_1,t_2} &= W_2(\tanh(W_1([\mathrm{h}_l^{t_1}; \mathrm{h}_v^{t_2}]) + b_1)) \\
\mathrm{s}_{l\to a}^{t_1,t_2} &= W_4(\tanh(W_3([\mathrm{h}_l^{t_1}; \mathrm{h}_a^{t_2}]) + b_3))
\end{aligned}$$

接下来使用 softmax 函数计算注意力权重 $w_{l\to v}$ 和 $w_{l\to a}$，并使用共享掩码屏蔽其他特征。

$$\begin{aligned}
\mathrm{w}_{l\to v}^{t_1,t_2} &= \frac{e^{\mathrm{s}_{l\to v}^{t_1,t_2}}}{\sum_{t_3=1}^{L} e^{\mathrm{s}_{l\to v}^{t_1,t_3}}} \\
\mathrm{w}_{l\to a}^{t_1,t_2} &= \frac{e^{\mathrm{s}_{l\to a}^{t_1,t_2}}}{\sum_{t_3=1}^{L} e^{\mathrm{s}_{l\to a}^{t_1,t_3}}} \\
\mathrm{w}_{l\to v} &= \mathrm{w}_{l\to v} \circ \mathrm{smask}_{l\to v} \\
\mathrm{w}_{l\to a} &= \mathrm{w}_{l\to a} \circ \mathrm{smask}_{l\to a}
\end{aligned}$$

获得非文本共享上下文向量:c_v，c_a，$\mathrm{c}_v, \mathrm{c}_a \in \mathbb{R}^{L\times d_h}$

$$\begin{aligned}
\mathrm{c}_v &= \mathrm{w}_{l\to v}\mathrm{h}_v \\
\mathrm{c}_a &= \mathrm{w}_{l\to a}\mathrm{h}_a
\end{aligned}$$

接着连接 c_v、c_a 和 h_l 并将其输入到融合 $LSTM$ 网络中，产生 $\mathrm{r}_s \in \mathbb{R}^{L\times 3d_h}$ 。然后进一步使用自注意力层，表示为 $SelfAttentionLayer$，来学习最终表示。自注意力层类似于跨模态注意力网络，使用 r_n 的最后一步表示作为共享表示 r_s。

$$\begin{aligned}
\mathrm{r}_m &= \mathrm{LSTM}_{\mathrm{fusion}}([\mathrm{c}_v; \mathrm{c}_a; \mathrm{h}_l]) \\
\mathrm{r}_n &= \mathrm{SelfAttentionLayer}(\mathrm{r}_m)
\end{aligned}$$

私有模块:

使用注意力网络来学习信息性和模态私有表示，使模型能够捕获非文本模态中包含的唯一信息，其中评分函数的参数:$W_5, W_6 \in \mathbb{R}^{d_h\times l}$, $b_5, b_6 \in \mathbb{R}$。

$$\begin{aligned}
\mathrm{s}_v^t &= W_5\mathrm{h}_v^t + b_5 \\
\mathrm{s}_a^t &= W_6\mathrm{h}_a^t + b_6
\end{aligned}$$

接着使用私有掩码来忽略其他特征并应用 softmax 函数来获得注意力权重。

$$\begin{aligned}
\mathrm{s}_v &= \mathrm{s}_v + (1 - pmask_v) * (-10^8) \\
\mathrm{s}_a &= \mathrm{s}_a + (1 - pmask_a) * (-10^8)
\end{aligned}$$

最后计算加权和并将它们表示为 p_v 和 p_a，称为私有表示。

$$\mathrm{w}_v^t = \frac{e^{\mathrm{s}_v^t}}{\sum_{t_1=1}^{L} e^{\mathrm{s}_v^{t_1}}}$$

$$\mathrm{w}_a^t = \frac{e^{\mathrm{s}_a^t}}{\sum_{t_1=1}^{L} e^{\mathrm{s}_a^{t_1}}}$$

$$p_v = \mathrm{w}_v \mathrm{h}_v$$

$$p_a = \mathrm{w}_a \mathrm{h}_a$$

回归层:

由具有 *ReLU* 激活函数的两层网络实现的回归层来融合共享和私有表示，其中 $W_f \in \mathbb{R}^{d_h \times 5d_h}$ ，$W_o \in \mathbb{R}^{1 \times d_h}$ ，$b_f \in \mathbb{R}_h^d$,$b_o \in \mathbb{R}$。

$$\hat{y} = W_o(\mathrm{ReLU}(W_f([r_s; p_v; p_a]) + b_f)) + b_o$$

在了解了模型的基本原理后，从论文所给代码库https://github.com/lzjjeff/TCSP.git下载代码，按 yml 文件搭建所给环境，阅读代码，了解每个文件的功能，下载数据集，尝试复现结果。

代码核心文件及函数：

run_tcp.py: 训练和评估一个包含翻译模型和回归模型的机器学习系统，基于多模态输入（如文字、视频、音频等）进行学习，实现了一个深度学习模型的训练、验证、测试流程，包括了翻译模型和回归模型的优化与评估。def information_entropy() 用来计算注意力权重的信息熵惩罚。信息熵（信息量）可以用来衡量模型预测的不确定性，进而提高模型对目标词和源词之间的区分能力。def train_translation() 函数用于训练翻译模型。每一个 epoch 中，它遍历训练数据并计算损失。def train_regression() 用于回归模型的训练，类似翻译模型训练，每个 batch 通过计算回归损失来优化模型。在每个 epoch 结束后，验证模型的表现。def test_regression() 用于测试回归模型阶段加载最佳模型并进行评估。输出测试集的损失、准确率、F1 得分、MAE（均方误差）和相关性（Correlation）等指标。

model.py: 实现了 TCSP 的框架，涉及多模态情感分析的神经网络架构，结合了文本、视觉和音频数据的交叉模态预测。Translation(): 负责文本编码（LSTM）和解码，并实现基于注意力机制的交互。通过 _get_attn_weight 计算注意力权重，编码输入序列并将其翻译为目标序列。CrossAttention(): 实现源模态与目标模态间的交叉模态注意力机制，输出模态间的语义交互特征。通过线性变换和 softmax 计算注意力权重。加权聚合源模态的隐藏表示，形成上下文向量。SoftAttention(): 用于单一模态的注意力计算，帮助提取重要的时间步特征。以加权求和的形式得到关键时间步的聚合表示。Classifier(): 基于提取的特征进行分类，构建一个两层全连接网络并添加 Dropout 正则化。Regression(): 多模态特征融合的核心部分，结合语言（w）、视觉（v）、音频（a）三种模态。使用三组 LSTM 编码器分别对多模态输入进行特征提取。使用 CrossAttention 模块计算视觉-文本、音频-文本的注意力机制。提供模态间的互补特性与一致性建模（如 w2v_comp_mask 和 w2a_comp_mask）

4.3.2 复现的难点以及细节

1。首先，最开始最基础也是最重要，遇到的最大的最难以解决的难点是配置实验环境，在此花了很大的功夫，在最开始搭建环境的时候，遇到用项目给的环境文件使用 conda env create -f environment.yml 创建虚拟环境的时候但一直失败，显示没有找到这些包，先尝试更改配置包的下载源，但仍然搜索不到或者下载缓慢。之后尝试一个一个手动 pip 对应版本的包，因为有些包会因为不同版本而出现冲突。经仔细一个个排查发现有些包只存在于 linux 系统架构下，并且我的电脑没有 GPU，只能用 CPU 跑，肯定非常慢，请教了助教学长后换到了云端 colab 进行搭建配置，成功。

2. 在复现论文的过程中，本文一共有两个数据集 mosi 和 mosei，因为 mosei 数据集很大，跑起来非常慢，于是尝试按比例减小训练集和回归集测试的 epoch。

4.3.3 提出的创新方法以及实现的细节

在数据处理阶段，即 dataset.py 中：

1. 减少多余的计算目前在 _getitem_ 和 collate_fn 中，数据的归一化和填充操作每次都进行，可以考虑将某些操作移到初始化时，或者避免重复计算。在 _getitem_ 中对音频和视频的归一化操作，每次都做一次，而不是在数据加载阶段做。使用 np.nan_to_num 来代替手动设置 np.isfinite，使得音频和视频数据的处理更简洁。通过字典提取数据并统一处理，减少了每次访问数据时的计算开销。

2. 减少内存使用在 collate_fn 中，创建了很多大的 Tensor（如 text_tensor, video_tensor,audio_tensor）。考虑只在需要时才创建并进行填充，减少内存消耗，尤其是在处理长序列时。采用 torch.zeros 而不是在每次循环中创建 Tensor，这样避免了多次重新分配内存。

3. 优化 sort_sequences 在排序过程中，使用 torch.Tensor 来存储长度，并进行排序。可以优化成只一次排序，减少计算量。直接返回排序后的数据。

4. 数据归一化改进在 _getitem_ 中，归一化操作可以通过使用 np.nan_to_num 或者 torch 来更简洁地处理。

在 model.py 中：

1. 在处理 LSTM 的输出时进行批量归一化是有益的，但可能会导致训练不稳定。尝试将批量归一化应用于中间层而非最终输出，将 self.bn 替换为 LayerNorm，这对于 RNN（LSTM）输出可能更稳定。

2. 在注意力机制初始化的时候，交叉注意力和软注意力中的权重初始化方式没有明确提及，尝试使用更标准的初始化方法（如 xavier_uniform 或 kaiming_uniform），以帮助模型更快收敛。

3. 尝试替换 LSTM 模块为 BERT 层，用于生成更高质量的上下文表示。在 BERT 之上添加专用的分类层和辅助任务层，实现目标边界检测与情感分类。使用 transformers 库中的 BertModel 类来加载预训练的 BERT 模型。使用 BERT 的输出作为特征输入到后续的分类层。

5. 实验 *ExperimentalResults*

5.1. A Unified Model for Opinion Target Extraction and Target Sentiment Prediction [8]

5.1.1 实验数据集

本实验使用的数据集来源于 SemEval 系列任务中的餐馆（restaurant）领域数据，包括 rest14、rest15、rest16 三个数据集。具体来说：

- rest14：来自 SemEval-2014 任务 4 餐馆评论数据。该数据集为最早期与目标级情感分析相关的标准数据集之一。
- rest15：来自 SemEval-2015 任务 12 中的餐馆评论数据集。该数据集在标注策略和情感定义上与 rest14 相似，但包含更为严格的任务定义和稍有差异的标注格式。
- rest16：来自 SemEval-2016 任务 5 中的餐馆评论数据集。该数据集整合前两年任务的经验，数据规模与分布略有不同，旨在进一步测试模型对新标注与细微差别的适应性。

在实验中，我们分别对上述三个数据集进行训练与测试，并在相同参数设定与模型结构下对比其最终结果，从而验证模型的稳健性与可迁移性。

5.1.2 实验设置

- 模型参数与优化策略：根据前期的超参数设定与作者建议，实验中对 rest14、rest15、rest16 数据集统一使用相同的参数配置和训练策略。例如，词向量维度 (dim_w) 设为 300，LSTM 隐状态维度 (dim_ote_h, dim_ts_h) 分别为 50，dropout 率设为 0.5。优化器选择 Adam，初始学习率为 0.001，并在训练过程中对学习率做简单衰减。此

Model	rest14	rest15	rest16
E2E-ABSA (论文数据)	0.6710	0.5727	0.6431
E2E-ABSA (复现数据)	0.6636	0.5338	0.6302

图 2. 结果比较

外，实验中严格使用 CPU 环境并设定确定的随机种子 (dynet_seed 与 random_seed) 以保证结果可重复性。在运行前根据 config.py 中的参数为每个数据集加载与其对应的默认配置，并在 main.py 中解析。

- 预训练词向量与情感词典：使用 GloVe 预训练词向量（如 glove.840B.300d.txt）初始化词嵌入，对不存在预训练向量的词用随机向量初始化。载入 mpqa_full.txt 情感词典，辅助建立情感感知语言建模特征，强化模型对上下文情感信息的建模能力。
- 训练与验证策略：对训练集进行若干轮 (epoch) 训练，每个 epoch 结束后在开发集 (验证集) 上评估模型性能（F1 值），若性能提升则保存模型。最终使用在验证集上表现最优的模型在测试集上进行评估并报告结果。

5.1.3 复现结果

为了便于比较，我运行了模型，并在 rest14、rest15、rest16 上报告了模型的结果。

如图 2所示，从结果可以看出，本次复现在三个数据集上的性能指标与原论文报告的数值非常接近。这说明在严格遵循原始代码、参数设置、数据处理流程和环境要求的条件下，本研究成功再现了 E2E-TBSA 模型的实验结果。同时，不同数据集上的结果也体现了模型在跨年度、标注差异略有变化的数据集上的稳健性。

5.2. Polarity and Intensity: the Two Aspects of Sentiment Analysis[14]

5.2.1 *unimodal*

如表5.2.1为单模态下不同任务的实验结果。相较原论文的效果还是略差，但是各个任务的趋势相同，比如在声音模态中 S+I+P 的皮尔逊相关系数最高，语言模态的各个任务的皮尔逊相关系数均高于其他两个模态。

CC	S	S+P	S+I	S+I+P
Vocal	0.12	0.145	0.11	0.152
Visual	0.09	0.11	0.115	0.103
Verbal	0.402	0.455	0.43	0.421
Human	0.820	-	-	-

5.2.2 *multimodal*

如表5.2.2为多模态下不同融合策略下的实验结果。与论文中所提一致。多模态模型从多任务学习中的收益较小。实际上，HF 和 LF 模型在使用单任务学习时表现更好。对于 TFN 模型，只有 S+P 模型（情感回归 + 极性分类）优于 S 模型（仅情感回归），但提升不显著。对于 EF 模型，多任务学习能够提升性能，这可能是因为 EF 模型在合并模态时没有偏见，各个特征对模型的影响更大。

CC	S	S+P	S+I	S+I+P
EF	0.471	0.470	0.465	0.481
TFN	0.446	0.461	0.445	0.430
LF	0.454	0.409	0.429	0.43
HF	0.470	0.422	0.460	0.423
Human	0.820	-	-	-

5.2.3 创新方法

虽然实验使用模型较多，但是不同的模型间其实差别并不太大，主要是输入与输出之间的区别，这里的优化选择多模态的 EF 融合策略的模型进行尝试。

在使用 Early Fusion 的多模态任务学习中，我尝试对现有的模型架构进行替换。当前模型使用的是单向 LSTM 层，由于使用的数据集是有时间序列的，所以尝试换为双向 LSTM 层，这样可以更好地捕捉前后依赖关系；目前使用的特征融合方式是简单的特征拼接（np.concatenate），更改为注意力机制，为每种模态特征赋予不同的权重，因为 Verbal 特征的效果很明显比其他两个模态的特征要更加有效，所以为 Verbal 赋予更大的权重。

更改后的实验效果如表5.2.3。创新后的结果反而没有原模型的效果好，尤其是在多任务的输出方面，皮尔逊相关系数远比原模型低。

CC	S	S+P	S+I	S+I+P
EF	0.471	0.470	0.465	0.481
EF(improved)	0.460	0.420	0.430	0.473

5.3. A Text-Centered Shared-Private Framework via Cross-Modal Prediction for Multimodal Sentiment Analysis[18]

5.3.1 实验数据集

$CMU-MOSI$: 包含 93 个 $YouTube$ 视频，总长 2 至 5 分钟，分割成 2199 个剪辑，每个剪辑都标注了从-3 强烈负面）到 3（强烈正面）的情感分数。[20]

$CMU-MOSEI$: 包含 23453 个视频话语，由 1000 名不同说话者关于 250 个主题的讨论组成，同样标注了情感分数。[1]

在实验中，利用了多种特征：使用 $GloVe$[11]嵌入表示文本，维度为 300；通过 $Facet$[5]提取视觉特征，捕捉 35 个面部动作单元，形成 35 维向量；以及通过 $COVAREP$ 提取声学特征 [4]，包括 12 个 Mel 频率倒谱系数，构成 74 维特征向量。这些特征共同支撑了模型的多模态情感分析任务。

5.3.2 实验设置

2 类准确度 (Acc)、$f1$ 分数 ($F1$)、平均绝对误差 (MAE) 和相关性 ($Corr$)。

Models	Parameters	MOSI	MOSEI
Cross-Modal Prediction	Batch Size	24	24
	Max Length	50	128
	Hidden Size	100	100
	Epochs	40	40
	Learning Rate	0.0001	0.0001
	Dropout	0.5	0.5
	Patience	5	10
Regression	Batch Size	24	24
	Max Length	50	128
	Hidden Size	100	100
	Epochs	30	30
	Learning Rate	0.001	0.001
	Dropout	0.5	0.5
	Selection Number	5	5
	Patience	5	5

图 3. 模型的超参数

MOSI	Acc	F1	MAE	Corr
原论文结果	80.9	81.0	0.908	0.710
trans_epoch=5，regre_epoch=30	76.8	77.0	0.976	0.649
trans_epoch=40，regre_epoch=30	79.1	79.2	0.914	0.697

图 4. 复现结果：MOSI

训练细节：

在最后一个线性层之前应用 $dropout$ 进行正则化，使用 $Adam$ 作为优化器，选择编号是所选共享/私有特征的数量 K_s 和 K_p 采用相同的值。

基线：

$EF-LSTM$，$LFLSTM$,MFN[21],$RAVEN$[17],MTN[12],$MulT$[15] 多模态路由[16]。

$TCSP(Base)$ 是本文的基础模型。模型架构与完整模型相同，但它不使用共享掩码和私有掩码。比较 $TCSP(Base)$ 和 $TCSP(Full)$，可以判断区分非文本模态的共享和私有特征是否有用。

5.3.3 复现结果以及创新结果分析

该论文在两个数据集上的复现结果如图 4和图 5所示。可以发现对于 MOSI：从 trans_epoch=5，regre_epoch=30 到 trans_epoch=40，re-

MOSEI	Acc	F1	MAE	Corr
原论文结果	82.8	82.6	0.576	0.715
trans_epoch=10，regre_epoch=5	80.2	80.3	0.607	0.687

图 5. 复现结果：MOSEI

gre_epoch=30，整体表现有所回升，尤其是 MAE 和 Accuracy 接近原论文的结果，但当参数和原文一样的时候，结果仍有些偏差，猜想是作者多次测验取了最好的一次。而对于 MOSEI 数据集，因为样本量足够大，模型对训练轮次的变化不如 MOSI 数据集敏感，尽管表现有所下降，但整体变化不大，训练过程更加稳定。

创新结果主要是优化了处理数据集的方法，在优化后运行时间缩短了 3 分钟。(也考虑网速，内存大小变化会影响运行时间，不过带来的结果也是明显的。)

6. 总结与展望

6.1. A Unified Model for Opinion Target Extraction and Target Sentiment Prediction [8]

本次复现实验对 E2E-TBSA 模型的训练流程、数据处理策略及参数配置进行了全面的对照与实现。在多个数据集（rest14、rest15、rest16）上的实验结果表明，所复现的模型性能与原报告数据基本一致，验证了方法的可重复性与结果的可比性。

目标级情感分析仍有众多可探讨空间。通过引入更为先进的预训练语言模型、加入多模态信息或改进模型结构（如引入图结构、关系推理）等手段，模型的上下文理解能力与领域适应性有望进一步提升。此外，考虑到现实应用场景中目标的多样化与数据分布偏差，对跨域迁移、低资源条件下的模型表现及对抗鲁棒性研究也将是未来的潜在研究方向。总体而言，在当前可重复性与性能指标的良好基础上，E2E-TBSA 任务仍具备丰富的拓展与优化潜力，为情感分析研究的深化与实际应用推广提供了广阔的发展空间。

6.2. Polarity and Intensity: the Two Aspects of Sentiment Analysis[14]

这篇论文提出了一种面向单模态与多模态场景的情感回归方法，系统实现了针对单任务与多任务的模型框架。这些模型充分利用了单模态（如文本、图像、音频）与多模态（如文本-图像、文本-音频）的情感特征，探究了不同融合策略下模型的准确度，通过设计高效的特征融合与任务协同机制，有效提升了情感回归的准确性和鲁棒性。

但尽管论文的方法取得了优异的实验结果，但仍存在以下可以进一步改进的方向。首先当前多模态情感回归主要依赖于特征拼接或注意力机制，但这些方法可能在处理高维模态数据时效率不足。未来可探索基于图神经网络（GNN）或动态特征选择的方法，进一步提高特征整合效率与效果。其次为了适应实时应用场景（如实时情感分析系统或人机交互），模型推理的延迟需进一步优化。未来可结合轻量化模型（如基于蒸馏或量化的模型）与动态推理策略，显著降低计算开销。最后，当前模型主要集中于学术数据集的验证，未来可以将方法扩展到实际场景中，如社交媒体情感监测、心理健康评估以及智能客服系统中，以验证其在真实场景中的性能与适用性。

6.3. A Text-Centered Shared-Private Framework via Cross-Modal Prediction for Multimodal Sentiment Analysis[18]

本次论文复现结果表明,训练轮次和数据集的丰富程度能够改善模型的准确性和情感细节捕捉能力。未来的工作可以着重于结合预训练模型（如 BERT、RoBERTa)提高模型对情感信息的理解,利用未标记数据通过半监督学习增强模型泛化能力，并探索新的特征融合策略，如注意力机制变体或 Transformer 架构，以提升多模态融合效果。此外，结合强化学习和迁移学习等方法，以及个性化情感分析的应用，将进一步推动情感分析领域的研究与实践发展。

参考文献

[1] 14

[2] A. Bagher Zadeh, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency. Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph. In I. Gurevych and Y. Miyao, editors, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2236–2246, Melbourne, Australia, July 2018. Association for Computational Linguistics. 2, 3

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. Cornell University - arXiv,Cornell University - arXiv, Sep 2014. 9

[4] G. Degottex, J. Kane, T. Drugman, T. Raitio, and S. Scherer. Covarep amp;x2014; a collaborative voice analysis repository for speech technologies. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2014. 14

[5] P. Ekman, W. V. Freisen, and S. Ancoli. Facial signs of emotional experience. Journal of Personality and Social Psychology, page 1125–1134, Jun 2006. 14

[6] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, page 1735–1780, Nov 1997. 9

[7] E. Kouloumpis, T. Wilson, and J. Moore. Twitter sentiment analysis: The good the bad and the omg! Proceedings of the International AAAI Conference on Web and Social Media, 5(1):538–541, Aug. 2021. 2

[8] X. Li, L. Bing, P. Li, and W. Lam. A unified model for opinion target extraction and target sentiment prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 6714–6721, July 2019. 3, 12, 15

[9] Z. Li, Y. Wei, Y. Zhang, X. Zhang, and X. Li. Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 4253–4260, 2019. 2

[10] M. Mitchell, J. Aguilar, T. Wilson, and B. Van Durme. Open domain targeted sentiment. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1643–1654, 2013. 2

[11] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Jan 2014. 14

[12] H. Pham, P. Liang, T. Manzini, L.-P. Morency, and B. Póczos. Found in translation: Learning robust joint representations by cyclic translations between modalities. Cornell University - arXiv,Cornell University - arXiv, Dec 2018. 14

[13] G. Qiu, B. Liu, J. Bu, and C. Chen. Opinion word expansion and target extraction through double propagation. volume 37, pages 9–27, 2011. 2

[14] L. Tian, C. Lai, and J. Moore. Polarity and intensity: the two aspects of sentiment analysis. In A. Zadeh, P. P. Liang, L.-P. Morency, S. Poria, E. Cambria, and S. Scherer, editors, Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML), pages 40–47, Melbourne, Australia, July 2018. Association for Computational Linguistics. 5, 13, 15

[15] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Jan 2019. 14

[16] Y.-H. H. Tsai, M. Ma, M. Yang, R. Salakhutdinov, and L.-P. Morency. Multimodal routing: Improving local and global interpretability of multimodal language analysis. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Jan 2020. 14

[17] Y. Wang, Y. Shen, Z. Liu, P. P. Liang, A. Zadeh, and L.-P. Morency. Words can shift: Dynamically adjusting word representations using nonverbal behaviors. Proceedings of the AAAI Conference on Artificial Intelligence, page 7216–7223, Aug 2019. 14

[18] Y. Wu, Z. Lin, Y. Zhao, B. Qin, and L.-N. Zhu. A text-centered shared-private framework via cross-modal prediction for multimodal sentiment analysis. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4730–4738, Online, Aug. 2021. Association for Computational Linguistics. 9, 14, 15

[19] H. Xu, B. Liu, L. Shu, and P. S. Yu. Double embeddings and cnn-based sequence labeling for aspect extraction. arXiv preprint arXiv:1805.04601, 2018. 2

[20] A. Zadeh. Micro-opinion sentiment intensity analysis and summarization in online videos. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, page 587–591, Nov 2015. 14

[21] A. Zadeh, P. P. Liang, N. Mazumder, S. Poria, E. Cambria, and L.-P. Morency. Memory fusion network for multi-view sequential learning. Proceedings of the AAAI Conference on Artificial Intelligence, Jun 2022. 14

[22] A. Zadeh, P. P. Liang, S. Poria, P. Vij, E. Cambria, and L. Morency. Multi-attention recurrent network for human communication comprehension. CoRR, abs/1802.00923, 2018. 9

7. 附件

A Unified Model for Opinion Target Extraction and Target Sentiment Prediction 复现的实验结果截图:

```
================================================================================
    -ds_name: rest14
    -dim_char: 10
    -dim_char_h: 50
    -dim_ote_h: 50
    -dim_ts_h: 50
    -input_win: 3
    -stm_win: 3
    -optimizer: sgd
    -n_epoch: 20
    -dropout: 0.5
    -emb_name: glove_840B
    -tagging_schema: BIEOS
    -rnn_type: LSTM
    -sgd_lr: 0.1
    -clip_grad: 5.0
    -lr_decay: 0.05
    -use_char: 0
    -epsilon: 0.5
    -dynet_seed: 1314159
    -random_seed: 1234
    -dim_w: 300
    -ote_tag_vocab: {'O': 0, 'B': 1, 'I': 2, 'E': 3, 'S': 4}
    -ts_tag_vocab: {'O': 0, 'B-POS': 1, 'I-POS': 2, 'E-POS': 3, 'S-POS': 4, 'B-NEG': 5, 'I-NEG': 6, 'E-NEG': 7,
Best results obtained at 19: ote f1: 0.8401, ts: precision: 0.6837, recall: 0.6446, ts micro-f1: 0.6636
Best model is saved at: ./predictions/rest14_0.663553.txt
================================================================================
```

图 6. rest14

```
==========================================================================================
    -ds_name: rest15
    -dim_char: 10
    -dim_char_h: 50
    -dim_ote_h: 50
    -dim_ts_h: 50
    -input_win: 3
    -stm_win: 3
    -optimizer: sgd
    -n_epoch: 25
    -dropout: 0.5
    -emb_name: glove_840B
    -tagging_schema: BIEOS
    -rnn_type: LSTM
    -sgd_lr: 0.1
    -clip_grad: 5.0
    -lr_decay: 0.05
    -use_char: 0
    -epsilon: 0.5
    -dynet_seed: 1314159
    -random_seed: 1234
    -dim_w: 300
    -ote_tag_vocab: {'O': 0, 'B': 1, 'I': 2, 'E': 3, 'S': 4}
    -ts_tag_vocab: {'O': 0, 'B-POS': 1, 'I-POS': 2, 'E-POS': 3, 'S-POS': 4, 'B-NEG': 5, 'I-NEG': 6, 'E-NEG': 7,
Best results obtained at 20: ote f1: 0.7000, ts: precision: 0.5493, recall: 0.5192, ts micro-f1: 0.5338
Best model is saved at: ./predictions/rest15_0.533785.txt
==========================================================================================
```

图 7. rest15

```
================================================================================
    -ds_name: rest16
    -dim_char: 10
    -dim_char_h: 50
    -dim_ote_h: 50
    -dim_ts_h: 50
    -input_win: 3
    -stm_win: 3
    -optimizer: sgd
    -n_epoch: 20
    -dropout: 0.5
    -emb_name: glove_840B
    -tagging_schema: BIEOS
    -rnn_type: LSTM
    -sgd_lr: 0.1
    -clip_grad: 5.0
    -lr_decay: 0.05
    -use_char: 0
    -epsilon: 0.5
    -dynet_seed: 1314159
    -random_seed: 1234
    -dim_w: 300
    -ote_tag_vocab: {'O': 0, 'B': 1, 'I': 2, 'E': 3, 'S': 4}
    -ts_tag_vocab: {'O': 0, 'B-POS': 1, 'I-POS': 2, 'E-POS': 3, 'S-POS': 4, 'B-NEG': 5, 'I-NEG': 6, 'E-NEG
Best results obtained at 18: ote f1: 0.7113, ts: precision: 0.6965, recall: 0.5756, ts micro-f1: 0.6302
Best model is saved at: ./predictions/rest16_0.630232.txt
================================================================================
```

图 8. rest16