实验五报告（多模态情感分类）

孙睿 10235304408 仓库地址：https://github.com/virtue-svg/Project5.git

1. 摘要（Abstract）

本实验围绕多模态情感分类任务，共分为**三个阶段**。在数据预处理后，**第一阶段构建基线模型（TF-IDF + ResNet18 concat）**，为后续提供参照，**第二阶段开展多模态对比与优化**。对比了 BERT+ResNet18（concat/gated/late）与预训练多模态模型（CLIP、BLIP），最终选择 CLIP。**第三阶段对CLIP进行优化**，进行网格搜索找到最优超参组合，并在附近进行微调，在分类头加入 LayerNorm 取得最优表现：定向增强前的验证集 Macro-F1 为 **0.6735**。进一步结合 bad case 定向增强与再训练后，验证集 Macro-F1 提升至 **0.8577**（基线 Macro-F1 为 **0.4118**），显著优于原始模型与对比模型。最终模型参数量约 **151.7M**，并输出测试集预测结果用于提交。实验表明，预训练多模态表征与数据驱动增强相结合是提升性能的关键。

2. 数据预处理

在实验开始前，我先对数据进行预处理，处理后的数据供三个阶段使用，具体处理如下：

1) 数据组织
做法：只以 `train.txt` / `test_without_label.txt` 中的 guid 为准，读取对应文本与图片；data 目录中未在列表内的文件视为冗余并忽略。

2) 文本清洗
基础清洗：去除 URL、@mention；统一空白；转小写；将 `#` 号处理为空格并保留词。
进阶清洗（轻量）：emoji 替换为 `<emoji>`；重复字符折叠（如 “soooo” → “soo”）；可选停用词过滤。
目的：降低噪声、保留情感线索，提高文本表示稳定性。

3) 图像处理
训练阶段：随机裁剪/缩放、水平翻转、轻微颜色抖动。
验证阶段：Resize + CenterCrop + 归一化（ImageNet 均值/方差）。
目的：增强模型鲁棒性，同时保证验证可复现。

4) 数据划分：固定随机种子，按标签比例进行分层划分（train/val）。

3. 阶段一：基线模型

3.1 目的

构建一个可运行、可复现的多模态基线流程，明确数据读取、训练与评估的完整链路，得到可对比的基准指标，为后续多模态对比与优化提供统一参照。

3.2 模型与方法

基线模型采用 **TF-IDF 文本特征**与 **ResNet18 图像特征**进行**拼接（concat）**，再通过 MLP 分类头输出三分类结果（positive/neutral/negative）。

选取理由：TF-IDF 作为稳健且计算代价低的文本表示，适合做基线；ResNet18 结构成熟、参数量适中；拼接融合简单直观，便于作为后续改进的对照基准。

损失函数为**带类别权重的交叉熵**，可以缓解标签分布不均衡带来的偏置；优化器使用 **AdamW**，兼顾收敛速度与泛化能力。

3.3 实验设置

- 固定划分：训练集3600条，验证集400条
- 固定随机种子
- 训练轮数：5
- 评估指标：Acc / Macro-F1 / Precision / Recall

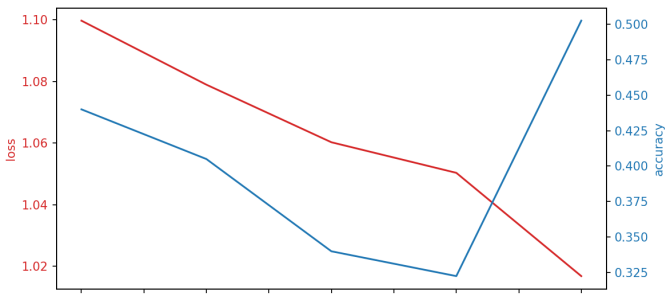
3.4 实验结果

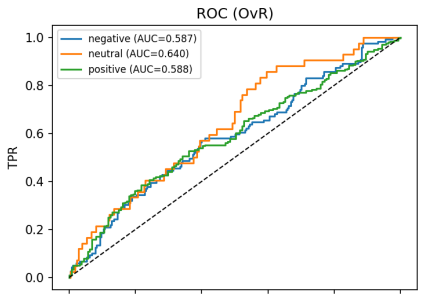
基线模型在第5轮取得最优结果，其各项指标如下：

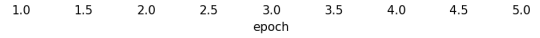
指标	Acc	Macro-F1	Precision	Recall
数值	0.5025	0.4118	0.4100	0.4207

分析： Acc 略高于 Macro-F1，说明少数类识别较弱；Precision 与 Recall 接近，模型未明显偏向某一类。ROC 曲线分离度一般，区分能力有限；训练/验证曲线趋于平稳，说明已基本收敛但提升空间不大。

以下为训练损失曲线和ROC曲线：







图表分析： 左图训练曲线后期趋于平稳，说明模型已基本收敛；右图 ROC 曲线未明显贴近左上角，区分能力一般，仍有提升空间。

3.5 消融实验

对于基线模型，我还简单地进行了消融实验，得到一个结果作为基准：

设定	仅文本	仅图像	多模态基线
Macro-F1	0.5267	0.3788	0.4118

分析：仅文本优于多模态基线，说明图像分支对情感分类贡献有限，简单拼接融合未发挥互补优势。

3.6 第一阶段问题与解决

1. **预训练权重缓存位置**：首次训练时权重默认下载到 C 盘导致空间紧张；通过设置缓存目录到 D 盘并清理旧缓存解决。
2. **数据冗余与对齐**：data 目录存在冗余文件，导致样本与标签不一致；改为严格以 `train.txt` / `test_without_label.txt` 的 guid 为准。
3. **PyTorch 导入 `WinError 1114`**：固定 `torch==2.1.2+cu121` 与 `numpy==1.26.4`。

---

4. 阶段二：多模态对比（选最优模型）

4.1 对比模型选择

本次我选择了以下模型进行对比：

- **BERT + ResNet18（concat / gated / late）**
  代表早期融合、门控融合、后融合三类结构。选择该组合是因为它是经典的"文本编码器 + 图像编码器"范式，结构清晰、成本可控，便于在固定其余变量的情况下比较不同融合位置对性能的影响。
- **CLIP / BLIP 代表预训练多模态模型。**加入它们是为了检验预训练多模态表征在本任务上是否优于轻量融合架构，并为最优模型选择提供更有说服力的对照。

参考论文：

- He, K., Zhang, X., Ren, S., Sun, J. **Deep Residual Learning for Image Recognition**. CVPR 2016. DOI: 10.1109/CVPR.2016.90.
- Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. arXiv:1810.04805.
- Radford, A. et al. **Learning Transferable Visual Models From Natural Language Supervision (CLIP)**. arXiv:2103.00020.
- Li, J. et al. **BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation**. arXiv:2201.12086.

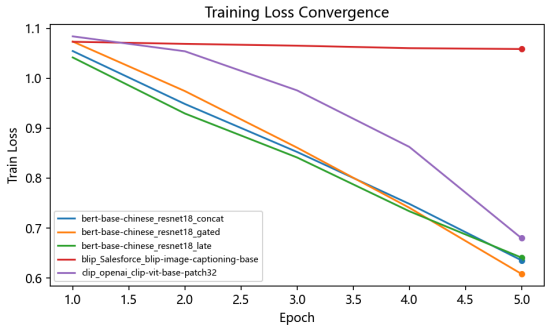
4.2 公平对比设置

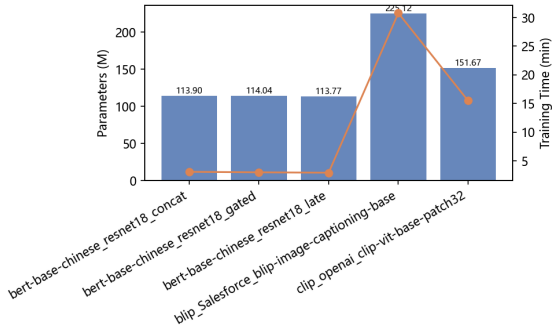
为保证对比具有可信度，我的设置如下：

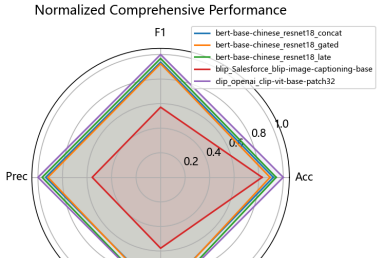
- 统一 train/val 划分
- 固定训练轮数：统一为 5 轮
- 固定 batch：BERT+ResNet18 与 CLIP 统一为 16；BLIP 因显存与模型规模限制采用 8（保持其余设置一致）
- 固定输入分辨率：图像统一到 224（由 ResNet/CLIP/BLIP 预处理器统一裁剪与归一化）
- 统一指标体系：Acc / Macro-F1 / Precision / Recall
- 仅改变模型结构，其余训练设置保持一致

4.3 对比结果

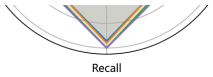
本次对比结果（验证集）可视化如下：







模型	参数量(M)	Acc	F1	Prec	Recall	训练时长(min)	训练轮数
ase-chinese_resnet1	113.90	0.6000	0.5425	0.5331	0.5725	3.10	5
ase-chinese_resnet1	114.04	0.5825	0.5354	0.5225	0.5823	3.01	5
base-chinese_resnet	113.77	0.6150	0.5600	0.5493	0.6065	2.95	5
force_blip-image-cap	225.12	0.5400	0.3303	0.3173	0.3505	30.79	5
penai_clip-vit-base-	151.67	0.6525	0.5785	0.5688	0.6069	15.50	5



图表分析： 结果表显示 CLIP 在 Acc 与 Macro-F1 上整体领先，其次为 BERT+ResNet18（late），BLIP 明显落后，支持选择 CLIP 作为阶段三优化对象。雷达图中 CLIP 在多数指标维度面积更大，说明综合表现更均衡；损失曲线显示 CLIP 收敛更稳定，部分融合模型波动较大；参数-时间图表明预训练模型训练成本更高，但性能收益更显著。
补充解释： CLIP 的图文对齐预训练更贴合跨模态分类任务，因此迁移效果更好；BLIP 该变体偏向生成/描述任务且部分参数需重新训练，表示稳定性较弱，因而表现最差。

4.4 第二阶段问题与解决

1. **CLIP 训练出现 CUDA 报错**（illegal instruction / CUBLAS）：将 `--num-workers` 设为 0，并降低 batch size，必要时用 `CUDA_LAUNCH_BLOCKING=1` 定位问题。
2. **BLIP 维度不匹配与 LazyModule 报错**：改用 LazyLinear，并在第一次前向后统计参数，避免未初始化参数调用 `numel()`。
3. **Windows 多进程 DataLoader 报错（pickle）**：避免在 DataLoader 中使用 lambda，改为可序列化的顶层函数，Windows 上优先用 `num-workers 0`。
4. **HuggingFace 缓存与权限警告**：提示开启开发者模式或以管理员运行以减少 symlink 警告（不影响训练）。

5. 阶段三：最优模型优化（CLIP）

5.1 优化目标与策略

- 目标：提升验证集 Macro-F1，保持训练稳定
- 策略：先进行超参网格搜索，得到最优超参后再进行一次微调与最终训练，之后再从数据出发，迭代改进模型
- 固定与阶段二相同的数据划分与随机种子，保证可比性

5.2 网格搜索

- 计划：对于超参组合进行网格搜索，每个超参有两个选项，共16组，根据Macro-F1选择最优超参组合。
- 搜索范围（共 16 组）：

超参	epochs	freeze-epochs	batch size	lr	weight decay	dropout
取值	8	1	4, 8	1e-5, 2e-5	0.0001, 0.0005	0.1, 0.2

- 最优超参组合：`bs8_lr1e-05_wd0.0001_dp0.2`，best_val_f1_macro = **0.6079**

5.3 微调与最终训练

- 计划：在最优组合附近继续微调，完成最终模型训练并保存权重
- 微调范围（围绕最优组合）：

超参	lr	weight decay	dropout	batch size	epochs	freeze-epochs
取值	8e-6, 1e-5, 1.2e-5	5e-5, 1e-4, 2e-4	0.15, 0.2, 0.25	8	8	1

- 微调最优：`bs8_lr8e-06_wd0.0001_dp0.15`，best_val_f1_macro = **0.6534**
  - 分析：较小学习率带来更稳定的微调更新；1e-4 的权重衰减与 0.15 的 dropout 在"正则化与表达能力"之间可能取得较好平衡，因此 Macro-F1 最优。
- 最终训练配置：epochs=12（实际训练到第 11 轮提前停止），batch=8，lr=8e-6，weight decay=1e-4，dropout=0.15，freeze-epochs=1，early-stop=3
- 最终训练结果：best_val_f1_macro = **0.6534**，训练时长约 **722.6s**

5.4 预处理改进对比

仅调整文本清洗与图像增强规则，其余训练设置不变，验证 Macro-F1 是否提升，作为数据层面的第一步改进。

具体调整说明：

- **baseline**（与 5.3 最优超参一致）：保留基础清洗（URL/@/空白/小写/# 处理），但不做 emoji 标记/重复字符折叠/停用词过滤；图像仅做标准 Resize+CenterCrop+归一化。
- **improved**：在 baseline 基础上额外开启 emoji 标记与重复字符折叠，并加入轻量图像增强（随机裁剪/翻转/轻微颜色抖动），其余超参与划分保持一致。
- 训练设置：统一 12 轮（early-stop=3），batch=8，lr=8e-6，weight decay=1e-4，dropout=0.15，freeze-epochs=1。

对比结果（12 轮，early-stop=3）：

设置	Acc	Macro-F1	Precision	Recall
baseline	0.7375	0.6534	0.7034	0.6300
improved	0.7375	0.6400	0.6541	0.6333

结论：本轮预处理改进未带来 Macro-F1 提升，说明在当前任务上"轻量清洗+增强"的增益有限，后续更侧重结构或训练策略优化。

增益有限的可能原因：

1. CLIP 已在大规模图文对齐数据上预训练，文本/图像本身具有较强鲁棒性，轻量清洗与增强提升空间有限。
2. 过度规整（emoji 统一、重复字符折叠）可能削弱部分情感强度信息，对少数类识别不利。

5.5 结构小改动（分类头）

在保持主干不变的前提下，仅对分类头做轻量调整（MLP / LayerNorm / 门控），控制变量观察结构微改是否带来收益。

具体调整说明：

- base：原始分类头（Linear → ReLU → Dropout → Linear）。
- ln：在融合特征后加入 LayerNorm，再进入同样的分类头。
- mlp：将分类头加深（两层隐藏 MLP），参数更多。
- gated：加入轻量门控（sigmoid gate）在图像/文本特征间做加权，再送入同样分类头。
- 其他设置全部固定：epochs=12、batch=8、lr=8e-6、weight decay=1e-4、dropout=0.15、freeze-epochs=1、early-stop=3。

对比结果（12 轮，early-stop=3）：

结构	Acc	Macro-F1	Precision	Recall
base	0.7375	0.6534	0.7034	0.6300
ln	0.7250	0.6735	0.6575	0.7110
mlp	0.6975	0.5900	0.5897	0.5904
gated	0.7025	0.6364	0.6326	0.6535

结果分析：

1. **ln 表现最好**：LayerNorm 稳定了融合特征的分布，减轻训练波动，提升对少数类的区分能力，因此 Macro-F1 明显提高。
2. **gated 次优**：门控能一定程度上抑制噪声模态，但仍受限于门控参数量较小，提升幅度有限。
3. **mlp 最差**：更深的头部增加了参数与优化难度，在数据规模有限的情况下易过拟合或梯度不稳定，导致 Macro-F1 明显下降。

结论：本轮中 **LayerNorm 头部（ln）** 获得最高 Macro-F1（0.6735），优于 base 与 gated；MLP 头部在当前设置下效果不佳。

5.6 最终模型选择

综合性能与稳定性，最终模型采用 **CLIP + LayerNorm 头部（ln）**，并沿用最优超参与统一训练策略作为阶段三的最终模型。

具体配置（横向表格）：

配置项	模型	头部	epochs	early-stop	freeze-epochs	batch size	lr	weight decay	dropout	max length	文本预处理	图像处理	num-workers
取值	openai/clip-vit-base-patch32	ln	12	3	1	8	8e-6	1e-4	0.15	64	基础清洗 + emoji 标记 + 重复字符折叠	Resize/CenterCrop/归一化	0

5.7 稳定性测试

我在相同配置下更换随机种子（seed=42/123/2024）重复训练，统计 Macro-F1 的均值与方差。

seed	42	123	2024
best_val_f1_macro	0.6735	0.6720	0.6750

均值 ± 标准差：0.6735 ± 0.0012
分析：多次复现波动极小，说明训练过程稳定，结果可信。

5.8 泛化测试

我在最终模型上进行验证集噪声/模糊扰动测试，观察鲁棒性下降幅度。

场景	Acc	Macro-F1	Precision	Recall
clean	0.7250	0.6735	0.6575	0.7110
noise	0.7250	0.6735	0.6575	0.7110
blur	0.7075	0.6657	0.6478	0.7194

分析：轻度噪声几乎不影响性能，说明模型对微小噪声较稳健；模糊会带来约 0.8 个 Macro-F1 的下降，提示图像清晰度对分类仍有一定影响，但整体降幅不大。

5.9 消融实验

在最终模型配置下，仅移除单一模态进行对比：

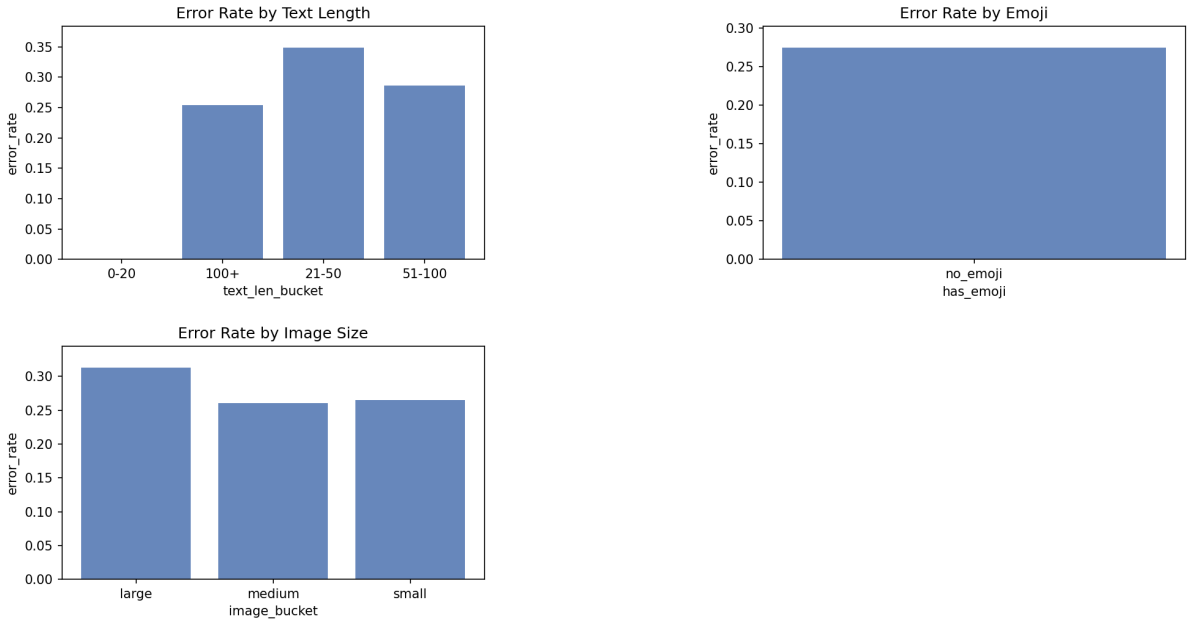
设置	Acc	Macro-F1
仅图像（去文本）	0.7050	0.5907
仅文本（去图像）	0.6875	0.6249

分析：去图像后 Macro-F1 仍高于去文本，说明文本分支贡献更大；但完整模型（0.6735）仍优于单模态，说明图像信息对部分样本存在补充价值，且完整模型发挥出了多模态融合的优势。

5.10 Bad case 挖掘与定向增强

Bad case 挖掘与归因：使用最终模型（CLIP + ln 头，best.pt）在验证集推理，筛出错误样本并生成 bad_cases.csv（含 guid、真实/预测、置信度、路径）。

错误类型分析（可视化）



简要解读：

1. 文本长度图：长文本段错误率更高，说明长文本噪声与主题漂移对情感判断有负面影响。
2. Emoji 图：含 emoji 的样本错误率偏高，提示情感符号带来更强语义波动。
3. 图像尺寸图：小尺寸图像错误率较高，说明图像信息不足时多模态融合收益下降。

定向数据增强：基于错误类型分析（长文本/含 emoji、小尺寸图像错误率更高），对 bad cases 执行增强：图像侧采用随机裁剪、水平翻转、轻微颜色抖动；文本侧做 emoji 标记与重复字符规范化，并生成增强文本表用于再训练。

5.11 再训练与预测

在定向数据增强完成后，我将增强样本与原训练集进行合并，保持最终模型超参不变，在合并训练集上再训练，最后使用增强后权重生成测试集提交文件。

增强后再训练结果（使用 batch=4 防止显存溢出）：

- best_val_f1_macro = **0.8577**
- 早停于第 10 轮

分析： 本次提升很大程度来自"增强样本并入训练集"带来的数据量与分布覆盖提升，因此 Macro-F1 的显著上升并非纯粹模型结构带来的改进，而是数据驱动增益的结果。

测试集预测说明： 使用增强后训练的最终模型对测试集进行推理，按 `guid,tag` 格式写出标签，生成最终预测结果。

至此，实验告一段落。

5.12 第三阶段问题与解决

1. **增强后再训练 CUDA 报错（CUBLAS）**：在合并增强样本后出现显存/内核错误，改用 `batch=4` 并保持其余超参不变后稳定训练完成。
2. **bad case 挖掘加载权重报错（layernorm keys）**：最终模型使用 ln 头部，加载时出现 key 不匹配；改为从权重中读取 `head_variant` 与 `dropout`，并使用 `strict=False` 解决。
3. **HuggingFace 下载超时**：出现 `ReadTimeout`，通过重试与使用本地缓存继续完成流程。

6. 思考与总结

收获：

1. 多模态任务中，预训练模型（CLIP）在对齐能力与迁移性能上显著优于轻量融合基线。

2. 结构微调（如 LayerNorm 头部）可稳定提升 Macro-F1，且训练代价低。
3. 数据驱动策略（bad case 定向增强）对性能提升贡献最大，但需谨慎保证对比公平性。（由于会将部分验证集并入训练集，可能会对其他测试有影响，例如消融实验）

局限性：

1. 增强后再训练的提升部分来自样本分布变化，难与早期对比实验直接公平比较。
2. 训练过程中存在偶发 CUDA 报错，影响完整性与效率。
3. 仍缺少更强的跨模态交互结构与更系统的损失函数对比。
4. 自我感觉最终各项指标水平可能不算优秀。

后续优化工作：

1. 引入跨模态注意力/轻量 Transformer 结构，增强早期融合效果。
2. 替换更强预训练骨干，例如使用更强CLIP变体。
3. 系统化比较 Focal/CB 等损失与组合损失的稳定性与收益。
4. 扩展数据清洗与弱标注修正，进一步减少噪声标签对训练的干扰。
5. 模型集成，使用2~3 个不同结构模型做投票/加权平均

感想： 作为最后一次作业，我把前四次作业积累的经验基本都用上了：网格搜索调优超参、模型对比可视化、再训练、稳定性/泛化测试，可以说是“集大成”的一次实验。虽然最终指标并未达到我期望的优秀水平（大模型建议 Macro-F1 需达到 0.75 以上才算优秀），但能将一学期所学的知识与技巧真正落地到本次实验中，我认为已经很有收获，也不是空洞的学习。