文本-语音双模态抑郁症检测模型

多媒体技术课程project

3220104147　余卓耘

目录
contents

01

选题背景及意义
(Background &Significance)

选题背景

传统临床抑郁症检测采用如脑电、量表等方式，复杂且成本较高；同时患者可能因担心偏见抗拒检测。近年来，自动抑郁症检测系统的研究逐渐兴起，可利用生活资料、访谈与心理咨询，辅助临床诊断诊断。

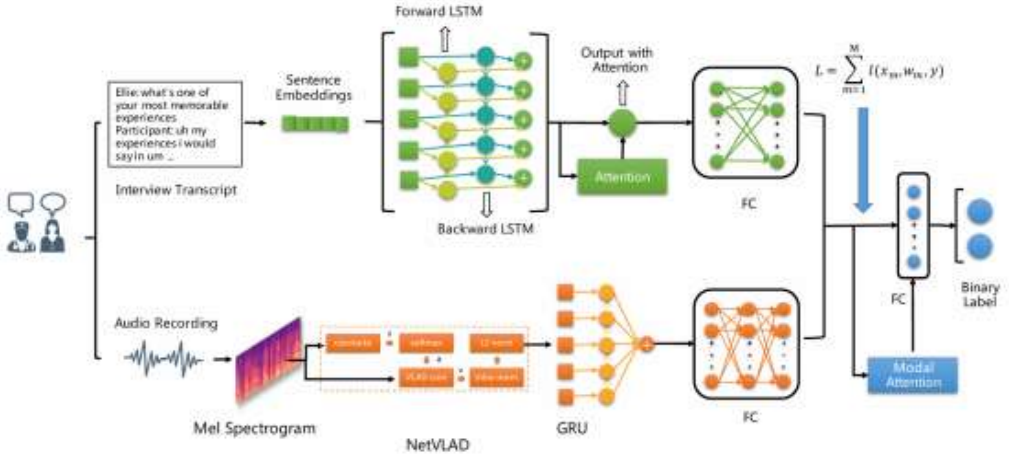
现有语音-文本检测方法

AUTOMATIC DEPRESSION DETECTION: AN EMOTIONAL AUDIO-TEXTUAL COR AND A GRU/BILSTM-BASED MODEL

*Ying Shen, Huiyu Yang, Lin Lin**

School of Software Engineering, Tongji University, P.R.China
{yingshen,2031552,1931542}@tongji.edu.cn

ICASSP 2022



项目简介

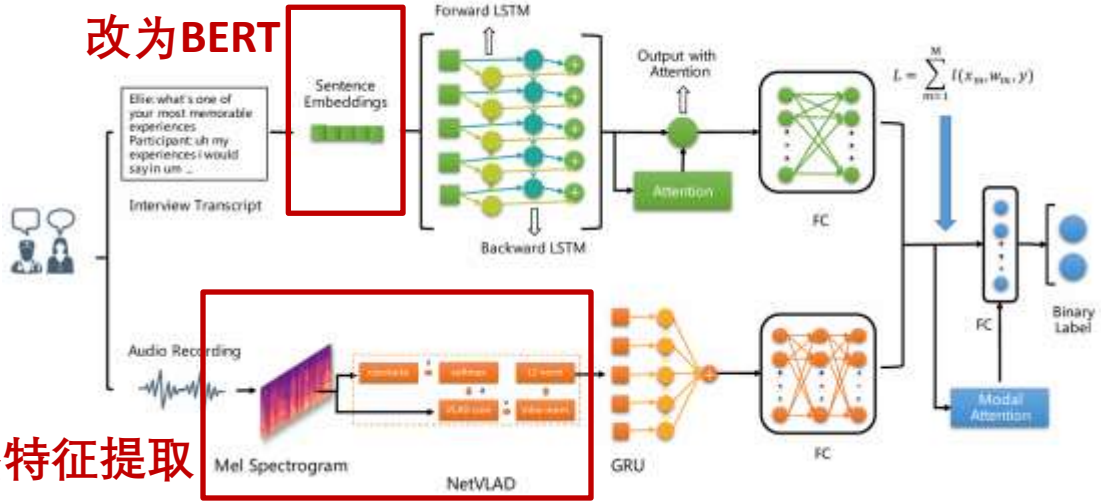
基于现有论文已开源的模型代码进行方法改良，旨在克服现有方法局限性，提升模型在语音-文本模态下抑郁症检测能力。

I.特征提取的改进

结合多个更基础(低维)的音频特征提取的文本特征，能够更全面地捕捉音频和文本中的信息，增进可解释性。

II.模型架构的优化

初步提取文本特征使用 BERT，相比原论文使用的更适用于语法分析的 ELMo，BERT 更强大，能够捕捉更复杂的语义信息。



开发环境

模型训练:
Google Colab平台
AutoDL算力平台 V100 16GB*1

版本管理:
Github

代码开发平台:
Vscode

运行环境:
目前Vscode+Colab 直接运行代码文件
→后续考虑利用colab-cli等工具开发CLI程序

工具包:

requirements.txt

```
...
torch==2.0.1
torchaudio==2.0.2
librosa==0.10.0
pandas==2.0.3
numpy==1.24.3
scikit-learn==1.3.0
transformers==4.31.0
kaldiio==2.17.2
scipy==1.10.1
tensorboard==2.13.0
matplotlib==3.7.2
soundfile==0.12.1
```

02

研究思路与方法

(Research Thoughts and Methods)

数据集（参考原论文）

EATD-Corpus数据集 EATD-Corpus Public

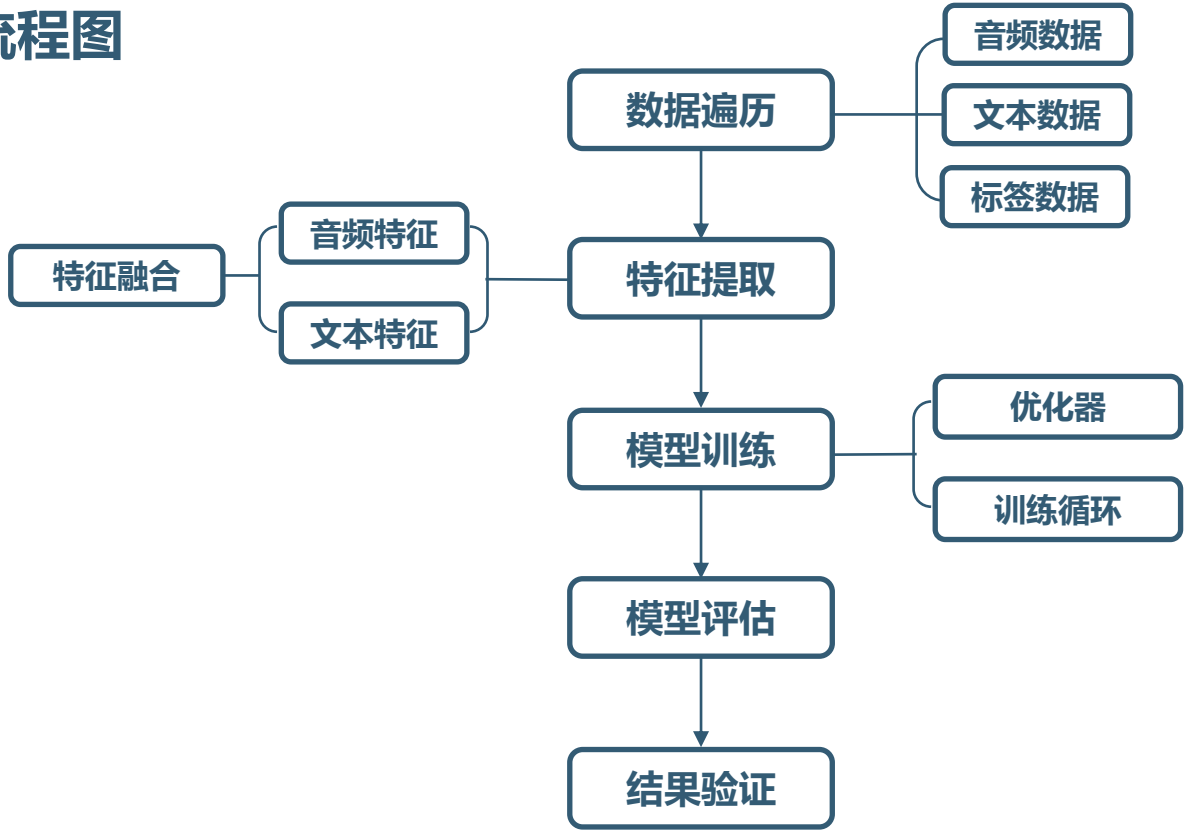
EATD（Emotion and Audio Text Dataset）是一个多模态数据集，主要用于抑郁症检测和情感分析任务。它结合了音频（WAV 文件）和文本（TXT 文件），旨在通过多模态特征（如语音、文本内容）来识别抑郁症相关的情感状态。它是第一个也是唯一一个包含中文音频和文本数据的公共抑郁症数据集。

数据集开源于Github: Fancy-Block/EATD-Corpus: An Emotional Audio-Textual Corpus

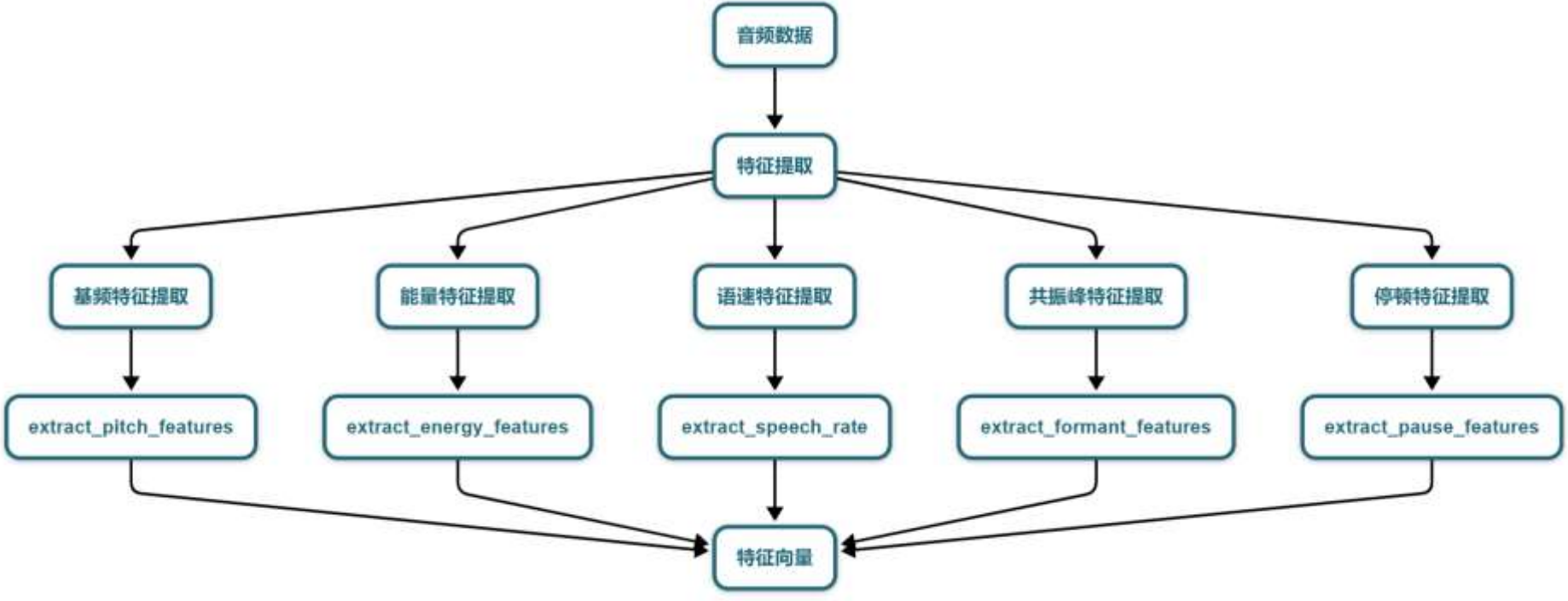
label.txt/new_label.txt：包含原始和标准化的SDS评分，用于评估志愿者的抑郁状态。
{positive/negative/neutral}.txt：文本文件，分别对应积极、消极、中性情感的文本。
{positive/negative/neutral}.wav：原始音频文件，记录志愿者的语音。

```
EATD-Corpus/
├── t_1/
│   ├── label.txt
│   ├── new_label.txt
│   ├── positive.txt
│   ├── positive.wav
│   ├── positive_out.wav
│   ├── neutral.txt
│   ├── neutral.wav
│   ├── neutral_out.wav
│   ├── negative.txt
│   ├── negative.wav
│   └── negative_out.wav
│
├── t_2/
│   ├── label.txt
│   ├── ...
└── ...
```

全流程图


数据遍历
音频数据
文本数据
标签数据
特征提取
音频特征
文本特征
特征融合
模型训练
优化器
训练循环
模型评估
结果验证

算法架构图——音频特征提取



实现细节——音频特征提取

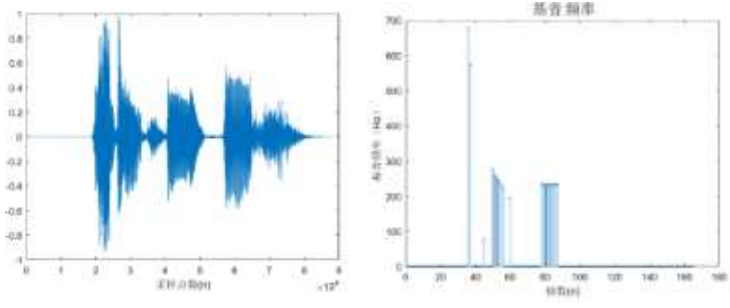
基频特征

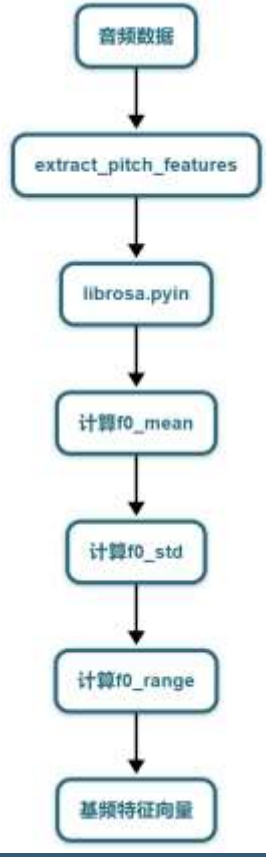
波形图表现：反映音频信号的频率变化。在波形图中，基频特征可以通过波形的周期性变化来观察。

提取方法：使用 librosa.pyin 函数提取基频特征，计算基频的均值（f0_mean）、标准差（f0_std）和范围（f0_range）。

抑郁症检测意义：基频特征可以反映说话者的语调和情绪状态，抑郁症患者通常语调较低。

```
函数 extract_pitch_features(音频数据, 采样率):
    // 使用PYIN算法提取基频
    基频, 浊音标志, 浊音概率 = librosa.pyin(
        音频数据,
        fmin=librosa.note_to_hz('C2'),
        fmax=librosa.note_to_hz('C7'),
        sr=采样率
    )

    // 计算统计特征
    基频均值 = np.nanmean(基频[浊音标志])
    基频标准差 = np.nanstd(基频[浊音标志])
    基频范围 = np.nanmax(基频[浊音标志]) - np.nanmin(基频[浊音标志])

    返回 [基频均值, 基频标准差, 基频范围]
```





实现细节——音频特征提取

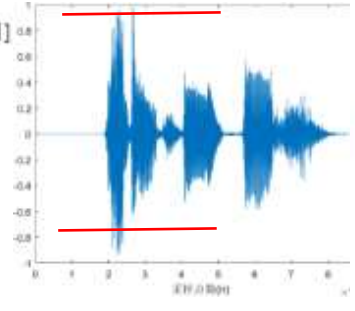
能量特征

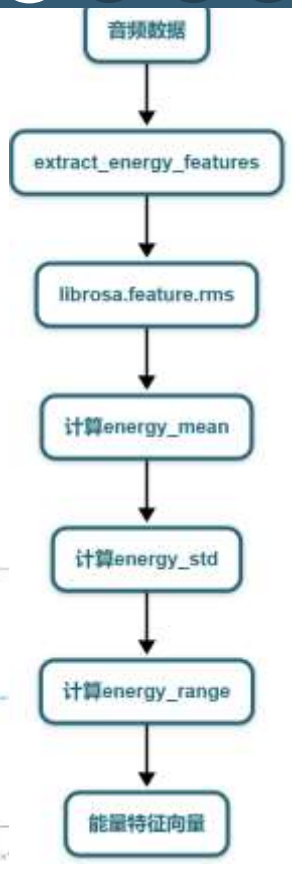
波形图表现：反映音频信号的振幅变化。在波形图中，能量特征可以通过波形的振幅大小来观察。

提取方法：使用 librosa.feature.rms 函数提取能量特征，计算能量的均值（energy_mean）、标准差（energy_std）和范围（energy_range）。

抑郁症检测意义：能量特征可以反映说话者的语音强度，抑郁症患者通常语音能量较低。

```
函数 extract_energy_features(音频数据, 采样率):
    // 计算短时能量
    帧长 = int(0.025 * 采样率)  // 25ms
    帧移 = int(0.010 * 采样率)  // 10ms
    能量 = librosa.feature.rms(
        y=音频数据,
        frame_length=帧长,
        hop_length=帧移
    )[0]

    // 计算统计特征
    能量均值 = np.mean(能量)
    能量标准差 = np.std(能量)
    能量范围 = np.max(能量) - np.min(能量)

    返回 [能量均值, 能量标准差, 能量范围]
```





实现细节——音频特征提取

语速特征&停顿特征

波形图表现：语速与停顿特征反映音频信号的节奏变化。在波形图中，语速与停顿特征可以通过波形的密集程度来观察。

提取方法：

语速特征：使用 librosa.onset.onset_strength 和 librosa.onset.onset_detect 函数提取语速特征，计算语速（speech_rate）。

停顿特征：计算短时能量，使用能量阈值检测停顿，计算停顿比例（pause_ratio）和停顿持续时间（pause_duration）。

抑郁症检测意义：可以反映说话者的说话速度，抑郁症患者通常语速较慢、停顿较多。

```
函数 extract_speech_rate(音频数据, 采样率):
    // 检测语音起始点
    起始强度 = librosa.onset.onset_strength(y=音频数据, sr=采样率)
    起始帧 = librosa.onset.onset_detect(onset_envelope=起始强度, sr=采样率)

    // 计算语速
    持续时间 = len(音频数据) / 采样率
    语速 = len(起始帧) / 持续时间

    返回 [语速]

函数 extract_pause_features(音频数据, 采样率):
    // 计算短时能量
    帧长 = int(0.025 * 采样率)
    帧移 = int(0.010 * 采样率)
    能量 = librosa.feature.rms(
        y=音频数据,
        frame_length=帧长,
        hop_length=帧移
    )[0]

    // 检测停顿
    阈值 = np.mean(能量) * 0.1
    停顿 = 能量 < 阈值

    // 计算停顿特征
    停顿比例 = np.sum(停顿) / len(停顿)
    停顿持续时间 = np.mean(np.diff(np.where(停顿)[0])) if np.sum(停顿) > 0 else 0

    返回 [停顿比例, 停顿持续时间]
```

实现细节——音频特征提取

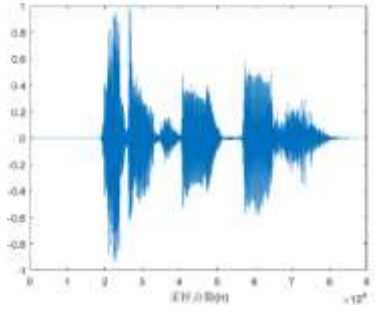
共振峰特征

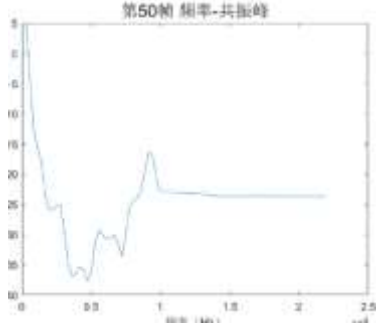
波形图表现：指在声音的频谱中能量相对集中的一些区域，其特征反映声道（共振腔）的物理特征、音频信号的频谱特性。在波形图中，共振峰特征可以通过频谱图的峰值来观察。

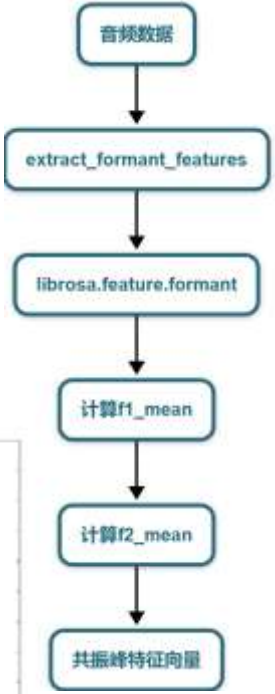
提取方法：使用 librosa.feature.formant 函数提取共振峰特征，计算共振峰的均值（f1_mean 和 f2_mean）。

抑郁症检测意义：共振峰特征可以反映说话者的语音质量和共鸣特性，抑郁症患者通常共振峰特性发生变化。

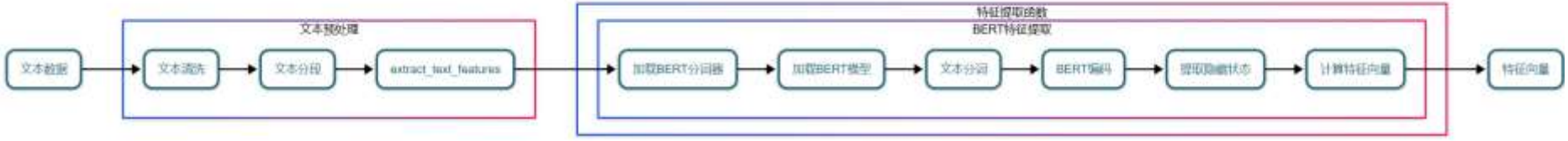
```
函数 extract_formant_features(音频数据, 采样率):
    // 提取共振峰
    共振峰 = librosa.feature.formant(y=音频数据, sr=采样率)

    // 计算前两个共振峰的统计特征
    第一共振峰均值 = np.mean(共振峰[0])
    第二共振峰均值 = np.mean(共振峰[1])

    返回 [第一共振峰均值, 第二共振峰均值]
```







算法流程图与实现——文本特征提取



1.使用 AutoTokenizer 和 AutoModel 从预训练的 BERT 模型（bert-base-chinese）中加载分词器和模型。

2.使用分词器对文本进行分词和编码。

3.将分词后的输入传递给 BERT 模型，获取模型的输出。模型输出的 last_hidden_state 是一个三维张量，表示每个 token 的嵌入表示。

4.通过取 last_hidden_state 的平均值（mean(dim=1)），将每个文本的特征压缩为一个固定长度的向量。

```
函数 extract_text_features(文本数据):
    // 加载BERT模型和分词器
    分词器 = AutoTokenizer.from_pretrained("bert-base-chinese")
    模型 = AutoModel.from_pretrained("bert-base-chinese")

    // 文本分词
    输入 = 分词器(
        文本数据,
        return_tensors="pt",
        padding=True,
        truncation=True
    )

    // BERT编码
    使用 torch.no_grad():
        输出 = 模型(输入)

    // 提取特征向量
    特征向量 = 输出.last_hidden_state.mean(dim=1).squeeze()

    返回 特征向量
```

隐藏特征提取与特征融合 具体实现（参考原论文）

文本特征提取

输入：文本特征序列（seq_len, batch, input_dim）。

使用**双向长短期记忆网络（BiLSTM）** 提取双向上下文特征。

使用 Attention 机制为每个时间步的特征分配权重。

输出：加权求和得到文本特征向量（batch, hidden_dim）

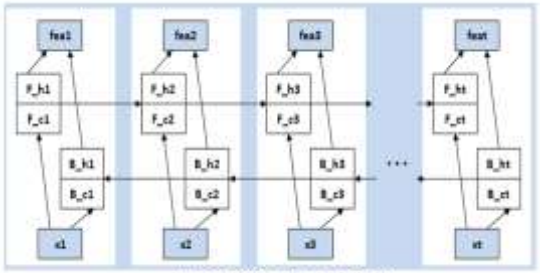


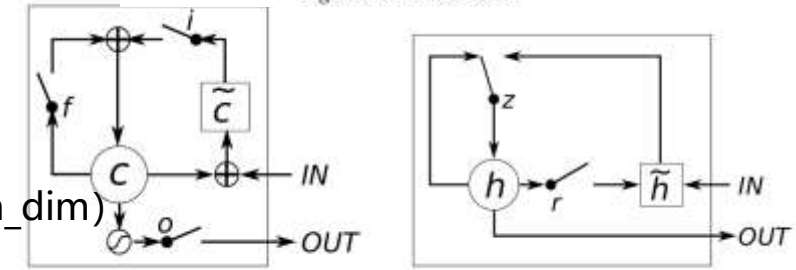
Fig. 2. Bidirectional LSTM

语音特征提取

输入：语音特征序列（seq_len, batch, input_dim）。

使用**门控循环单元（GRU）**提取语音特征。

输出：取最后一层隐藏状态作为语音特征向量（batch, hidden_dim）



(a) Long Short-Term Memory (b) Gated Recurrent Unit

多模态融合

将低级、高级的特征拼接在一起，将文本特征和语音特征拼接在一起，作为最终的输入特征。

使用全连接层进行分类，输出抑郁概率（0-1之间）。

模型训练 具体实现（参考原论文）

初始化模型和优化器

定义 FusionModel，包括文本和语音特征提取模块以及多模态融合模块。

使用 BCELoss 作为损失函数，计算预测值与真实标签之间的误差。

使用 Adam 优化器，学习率为 1e-4。

训练循环

每个 epoch 遍历训练数据集。

输入音频和文本特征，模型输出预测值。

计算损失并进行反向传播，更新模型参数。

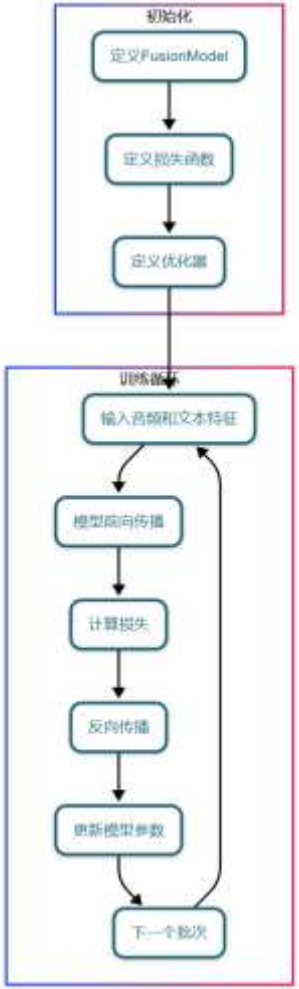
评估模型

在每个 epoch 结束后，评估模型在训练集上的性能。

打印当前 epoch 的损失值。

最终在33 epoch左右已经收敛，停止训练。

```
Train Epoch: 1 [14848/60000 (25%)]      Loss: 0.273648
Train Epoch: 1 [30208/60000 (50%)]      Loss: 0.098542
Train Epoch: 1 [45568/60000 (75%)]      Loss: 0.083506

Test set: Average loss: 0.0789, Accuracy: 9752/10000 (98%
```



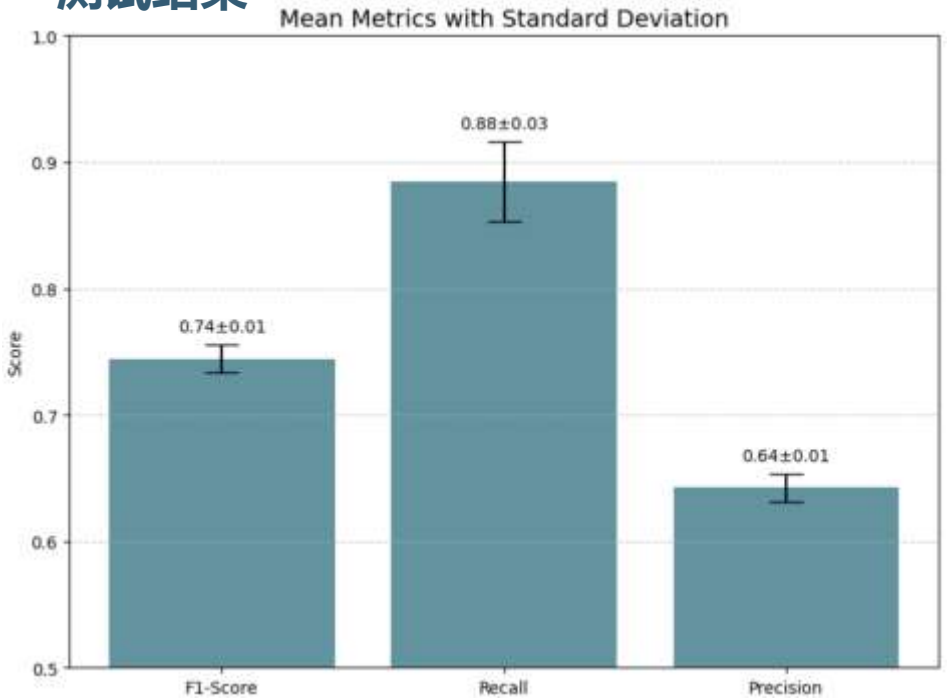
03

研究成果与效果展示
(Achievements)

评估指标

与原论文一致，使用病例检测的benchmark： **F1 Score、召回率（Recall）、精确率（Precision）**

测试结果



原论文结果

Table 4. Results of Experiments on EATD-Corpus

Features	Models	F1 Score	Recall	Precision
Audio	Multi-modal LSTM [13]	0.49	0.56	0.44
	SVM	0.46	0.41	0.54
	RF	0.50	0.53	0.48
	Decision Tree	0.45	0.44	0.47
	Proposed GRU model	**0.66**	**0.78**	**0.57**
Text	Multi-modal LSTM [13]	0.57	0.63	0.53
	SVM	0.64	**1.00**	0.48
	RF	0.57	0.53	0.61
	Decision Tree	0.49	0.43	0.59
	Proposed BiLSTM model	**0.65**	0.66	**0.65**
Fusion	Multi-modal LSTM [13]	0.57	0.67	0.49
	Proposed fusion model	**0.71**	**0.84**	**0.62**

消融实验结果

在改动的两个模块里删去其中一个后重新测试，发现去掉任何一个模块，检测能力都会下降。
改用Bert模块提取文本特征、用更基础低维的算法(Basic Extract)提取音频特征在该数据集上对提升模型性能均有效。（除了改动的两个模块其余与原论文红框圈出的部分一致）

测试结果

Features	F1 Score	Recall	Precision
Fusion with Mel Spec. & NetVLAD and Bert	0.72±0.02	0.86±0.02	0.62±0.02
Fusion with ELmo and Basic extract	0.72±0.03	0.85±0.02	0.63±0.01
Fusion with Bert and Basic extract	**0.74±0.01**	**0.88±0.03**	**0.64±0.01**

原论文结果

Table 4. Results of Experiments on EATD-Corpus

Features	Models	F1 Score	Recall	Precision
Audio	Multi-modal LSTM [13]	0.49	0.56	0.44
	SVM	0.46	0.41	0.54
	RF	0.50	0.53	0.48
	Decision Tree	0.45	0.44	0.47
	Proposed GRU model	**0.66**	**0.78**	**0.57**
Text	Multi-modal LSTM [13]	0.57	0.63	0.53
	SVM	0.64	**1.00**	0.48
	RF	0.57	0.53	0.61
	Decision Tree	0.49	0.43	0.59
	Proposed BiLSTM model	**0.65**	0.66	**0.65**
Fusion	Multi-modal LSTM [13]	0.57	0.67	0.49
	Proposed fusion model	**0.71**	**0.84**	**0.62**

04

相关思考与展望
(Comments and Expectations)

结果分析与可能的优化思路

1.**平均效果提升3~5%**：结合更低级的音频特征和BERT提取的文本特征，能够更全面地捕捉音频和文本中的信息，增加可解释性。推测因数据集较小(仅276组多模态数据)，高维隐藏特征提取效果不明显，**更低级特征的简单性和解释性使其在小数据集上表现更好。**原论文图表似乎也有体现。但由于无权限获得原论文使用的较大数据集，无法验证。

Table 3. Results of Experiments on DAIC-WoZ dataset

Features	Models	F1 Score	Recall	Precision
Audio	Gaussian Staircase Model [11]	0.57	-	-
	DepAudioNet [14]	0.52	1.00	0.35
	Multi-modal LSTM [13]	0.63	0.56	**0.71**
	SVM	0.40	0.50	0.33
	Decision Tree	0.57	0.50	0.57
	Proposed GRU model	**0.77**	**1.00**	0.63
Text	Multi-modal LSTM [13]	0.67	0.80	0.57
	Cascade Random Forest [8]	0.55	**0.89**	0.40
	Gaussian Staircase Model [11]	0.84	-	-
	SVM	0.53	0.42	0.71
	Decision Tree	0.50	0.67	0.40
	Proposed BiLSTM model	**0.83**	0.83	**0.83**
Fusion	Multi-modal LSTM [13]	0.77	0.83	0.71
	Proposed fusion model	**0.85**	**0.92**	**0.79**

Table 4. Results of Experiments on EATD-Corpus

Features	Models	F1 Score	Recall	Precision
Audio	Multi-modal LSTM [13]	0.49	0.56	0.44
	SVM	0.46	0.41	0.54
	RF	0.50	0.53	0.48
	Decision Tree	0.45	0.44	0.47
	Proposed GRU model	**0.66**	**0.78**	**0.57**
Text	Multi-modal LSTM [13]	0.57	0.63	0.53
	SVM	0.64	**1.00**	0.48
	RF	0.57	0.53	0.61
	Decision Tree	0.49	0.43	0.59
	Proposed BiLSTM model	**0.65**	0.66	**0.65**
Fusion	Multi-modal LSTM [13]	0.57	0.67	0.49
	Proposed fusion model	**0.71**	**0.84**	**0.62**

2. **召回率提高相对多,精确率提升相对少**：召回率的提高通常意味着模型能够更好地识别正例，而精确率关注的是模型预测为正例的样本中实际为正例的比例。可能是因为模型在正例和负例上的预测较为平衡，或者模型在负例上的预测仍然较为保守。但是，由于数据较少，提升幅度有限且实验存在误差，结果有待进一步验证。

→调整分类阈值、改进特征选择、优化模型结构与模型参数

参考文献

数据集

- EADT-Corpus：Fancy-Block. (n.d.). https://github.com/Fancy-Block/EATD-Corpus

代码库

- Transformers：https://github.com/huggingface/transformers
- BiLSTM：Karpathy, A. (2015). https://karpathy.github.io/2015/05/21/rnn-effectiveness/
- GRU：Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). https://arxiv.org/abs/1412.3555

论文

- Automatic Depression Detection：Shen, Y., Yang, H., & Lin, L. (2022). AUTOMATIC DEPRESSION DETECTION: AN EMOTIONAL AUDIO-TEXTUAL CORPUS AND A GRU/BILSTM-BASED MODEL. arXiv:2202.08210. https://arxiv.org/abs/2202.08210 (ICASSP 2022)

Thanks

答辩人：余卓耘