音频模态：转写、结构化信息与情绪变量制作说明

多模态信息结构化提取与情绪变量构造系统 · 按当前代码整理

> **先回答一个问题：一段录音里说了哪些事实？说话内容表达的情绪，和声音本身听起来的情绪是否一致？**

一、理解版：从问题到结果

音频里其实藏着两层信息：第一层是“说了什么”，例如人物、事件和时间；第二层是“怎么说的”，例如声音听起来放松、急促还是压抑。系统把这两层分开记录，就像采访时一边记下受访者的原话，一边在旁边记下他的语气。

不同录音的格式就像大小、速度都不同的磁带，不能直接用同一把尺子测量。系统先把它们统一成同一种声音格式，再开始分析。这样无论上传的是普通录音还是压缩音频，后面的步骤都能用相同方式处理。

长录音中有说话、停顿和完全安静的地方。系统像一个能听见“开口声”的小助手，先找到有人说话的区间，再在停顿附近剪开。每一小段都保留原来的起止时间，方便研究者回听和复核。

第一位助手负责“听写”：把每个小片听成文字。第二位助手阅读这些文字，用固定的问题整理出摘要、人物、事件、时间、地点、风险线索，并判断文字表达出的情绪。这样，研究者可以直接在表格里搜索和统计，而不必反复播放录音。

还有一位助手完全不看文字，只听声音本身：音调高不高、说得快不快、停顿多不多、声音有没有明显起伏。它给出一个声音情绪和一个可信程度分数。最后把“文字看起来的情绪”和“声音听起来的情绪”并排，看看两者是否一致。

> **边界提醒：**音频项目的 Qwen 只处理 ASR 文本，Emotion2Vec 才直接处理声音。服务器内存较小时应保持串行；若本地模型不可用，程序可能进入 mock 或 API 模式，必须从输出字段确认实际模式。

二、学术版：可直接放入方法部分的表述

> 音频经 ffprobe 预检并由 ffmpeg 规范化为单声道 16 kHz PCM WAV，随后以 FunASR FSMN-VAD 识别语音活动区间，在 30 秒目标长度和 8–45 秒边界约束下生成片段；使用 Fun-ASR-Nano-2512（language=zh，itn=True）进行自动语音识别，将 transcript 输入 Qwen qwen3.6-plus 完成结构化字段与语义情绪提取，并以 Emotion2Vec 识别声学情绪；融合阶段保留可用的声学标签，并构造文本情绪与声学情绪的一致性指标。

三、实际使用的包、模型与职责

组件/模型	作用	本项目实际怎么用
ffprobe / ffmpeg	格式标准化与切片	读取时长、复制非 ASCII 路径、输出 mono/16 kHz/PCM WAV，在 VAD 区间写出音频片段。
FunASR FSMN-VAD	语音活动检测	识别说话起止毫秒；VAD 不产生 transcript。失败时才使用固定时长切片。
Fun-ASR-Nano-2512	自动语音识别（ASR）	本地 FunASR AutoModel 对片段 generate，language=zh、itn=True，得到 transcript。
Qwen qwen3.6-plus	文本结构化与语义情绪	读取 transcript，按 JSON 模板输出摘要、实体、事件、风险和 emotion_label。
Emotion2Vec base_finetuned	声学情绪识别	通过 FunASR AutoModel 本地推理，或在配置 token 时调用 ModelScope API；输出 label、score、topk。
Python csv / json / tqdm	批处理、断点与结果表	逐片段保存 JSON、CSV、progress.json，并支持失败重试和断点续跑。

四、按代码复现：每一步输入什么、做什么、得到什么

步骤	人话说明与复现动作	得到什么
1. 准备输入	上传 .wav、.mp3、.m4a、.flac 或 .ogg。当前网站为了低内存稳定性限制单个音频任务最多 3 个文件，并串行处理。	原始音频
2. 检查并统一格式	ffprobe 读取时长；ffmpeg 输出单声道 16 kHz PCM WAV。中文路径会先复制到 audio_ascii_cache。	标准 WAV
3. VAD 切片	FunASR FSMN-VAD 给出毫秒区间；merge_vad_ranges 合并到目标 30 秒，最短 8 秒、最长 45 秒。没有有效区间时使用固定时长回退。	`audio_segments/<source>/*_segXXXX.wav`
4. ASR 转写	Fun-ASR-Nano-2512 的 AutoModel.generate 读取每个片段，language=zh、itn=True；返回文本写入 transcript。	`audio_variables.csv` 中的 `transcript`
5. Qwen 结构化	Qwen 只接收 transcript 文本，不直接听音频；按模板提取 summary、entities、events、emotion_cues、emotion_label、time、location、risk_tags。	`audio_variables.csv`

步骤	人话说明与复现动作	得到什么
6. 声学情绪	Emotion2Vec 对 WAV 片段进行 utterance 级分类；score 是最高类别的模型分数，execution_mode 记录 local/api/mock。	`audio_emotion_variables.csv`
7. 融合与核对	可用声学标签优先写入 final_emotion_label；compare_emotion_consistency.py 另写 emotion_match=1/0。	`audio_fusion_variables.csv` / `audio_emotion_consistency.csv`

五、公式、权重与融合规则

$$label_i = \arg\max_k p_{i,k} \qquad score_i = \max_k p_{i,k}$$

$$EmotionMatch_i = \mathbb{1}(\text{TextLabel}_i = \text{AudioLabel}_i)$$

LaTeX 写法（可复制到论文编辑器）

```
label_i = \underset{k}{\operatorname{arg\,max}}\;p_{i,k}, \qquad score_i = \max_k p_{i,k}

\operatorname{EmotionMatch}_i = \mathbb{1}(TextLabel_i=AudioLabel_i)

\operatorname{ConsistencyRate}=\frac{\sum_{i=1}^{N}\operatorname{EmotionMatch}_i}{N}
```

声学最高类别： 若 Emotion2Vec 输出各类别分数 $p_{i,k}$，代码保存 $label_i=\arg\max_k p_{i,k}$，并保存 $score_i=\max_k p_{i,k}$。

文本/声学一致性： $EmotionMatch_i=\mathbb{I}(TextLabel_i=AudioLabel_i)$；样本一致率为 $\Sigma EmotionMatch_i/N$。当前融合不是加权平均，而是“声学标签可用则优先，否则回退到 Qwen 文本标签”。

公式中的英文变量名与实际输出列保持一致；若某一步是规则优先级而不是连续加权，文中已明确标出。

六、关键输出变量：英文名与中文解释

英文变量名	中文解释
`source_audio_id`	原始音频编号
`segment_id / segment_index`	片段编号及在原音频中的顺序
`segment_start_sec / segment_end_sec`	片段起止秒数
`transcript`	Fun-ASR-Nano-2512 生成的语音转写文本
`summary`	Qwen 根据 transcript 生成的一句话摘要

英文变量名	中文解释
`entities_raw / keywords_raw / events_raw`	实体、关键词、事件列表
`emotion_cues_raw`	文本中支持情绪判断的线索
`emotion_label`	Qwen 根据转写文本判断的语义情绪
`label`	Emotion2Vec 判断的声学情绪标签
`score`	Emotion2Vec 最高类别的模型分数，不等同于人工校准概率
`execution_mode`	声学模型实际运行方式：local、api 或 mock
`final_emotion_label`	融合后的标签；当前规则优先声学结果
`emotion_match`	文本情绪与声学情绪相同记 1，否则记 0

七、结果文件与研究用途

结果文件	用途
`audio_variables.csv`	转写、结构化字段和 Qwen 语义情绪
`audio_emotion_variables.csv`	Emotion2Vec 的 label、score、运行模式
`audio_fusion_variables.csv`	文本情绪、声学情绪和 final_emotion_label
`audio_emotion_consistency.csv`	文本/声学标签的 0/1 一致性
`audio_*_run_summary.json`	总数、成功数、失败数、跳过数和重试数

可以把 transcript 用于事件、人物、时间和风险变量，把 label/score 用于声学情绪强度或类别比较，再用 emotion_match 检查“说的内容”和“说话方式”是否一致。研究报告中应同时报告 execution_mode，不能把 mock 结果当作真实模型结果。

本篇说明作为项目复现参考，更多字段、原始响应和运行日志请结合下载结果中的 JSON、CSV、progress.json 与 config_snapshot.json 一起核对。