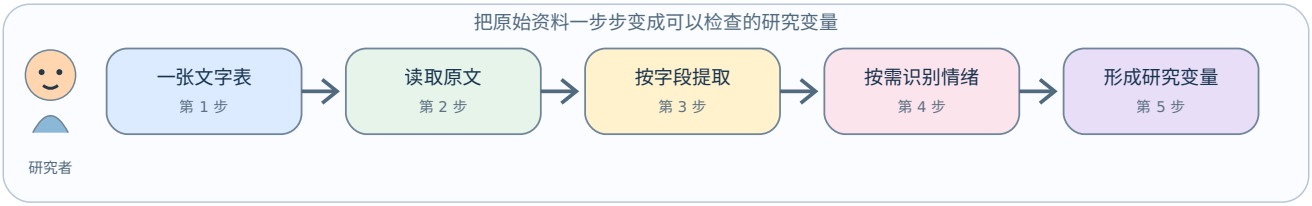
文本模态：多维情绪分数与结构化结果制作说明

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：一列新闻、年报或访谈文本，怎样变成可检查、可比较、可进入回归的情绪变量？**

一、理解版：从问题到结果



一篇新闻、年报或访谈原文，不能直接作为一个数字放进统计模型。这个项目做的事情，可以想成给每段文字准备了七把不同的尺子：一把量乐观，一把量悲观，一把量焦虑，其他尺子量愤怒、预期、中性和不确定。最后每篇文字都有一组分数，而不是只有一个简单的“好/坏”。

程序先像一位表格管理员一样，把逗号分隔表格或电子表格中的文字逐行读出来，检查每一行是否真的有内容，并记住它来自哪个文件、哪一行。这样研究者拿到结果后，能从任何一个数字追溯回原句。

长文章像一幅很长的画卷，一次摊在桌上可能看不全。只有当一条原文超过“每段最长字符数”时，系统才按句号、问号、感叹号、分号和换行优先切开，再把相邻的短句尽量组合到字符上限以内。特别长的一句话没有这些边界时，会按字符硬切；当前代码不会额外把逗号或冒号当作分段边界。

每个小段分别用七把尺子测量。实际任务中，StructBERT 会先对全部片段计算七维情绪分数；随后，如果用户填写了结构化字段，Qwen 再逐片提取字段。两条分析线使用同一批片段，情绪结果不会作为 Qwen 的输入，Qwen 的结构化结果也不会作为 StructBERT 的输入。整篇文章的情绪分数不是简单地数小段数量，而是让字数更多的小段承担更大的影响。

如果用户填写了结构化字段，系统会把同一段原文交给 Qwen，按字段提取公司名称、股票代码、产品名称等可核对信息；不填写结构化字段就不生成结构化结果文件。如果情绪选项填写“无”，这条文字就不会进入情绪模型，也不会生成悲观、焦虑等情绪列。只有明确需要情绪时，才额外运行本地模型。

> **边界提醒：**只有填写结构化字段才会生成 structured_results.csv；结构化字段依赖本次任务提交的 Qwen API Key，结果快照只记录是否配置，不保存密钥。情绪选项填写“无”时不会生成 emotion_results.csv；不要把缺少结果文件误认为任务失败。需要情绪时，StructBERT 的分数是模型输出，不等同于人工标注概率，正式研究前应抽样复核。

二、学术版：可直接放入方法部分的表述

> 文本由 pandas/openpyxl 读取并完成 text 列校验；当用户填写结构化字段时，原文逐行按用户分段输入 Qwen qwen-plus，按指定字段严格返回 JSON，并将字段展开为 CSV 列；当情绪选项明确为“无”时跳过 StructBERT 情绪推理，不生成情绪分数。需要情绪时，长文本依据句末边界和 tokenizer 的最大长度约束分块，每个文本块输入项目内 StructBERT 多标签模型，分类头输出 logits 并

> 经 sigmoid 得到 optimism、pessimism、anxiety、anger、expectation、neutral 和 uncertainty 等维度分数，再按有效字符长度加权聚合，并依据 label_config.json 中的阈值和权重生成 top_labels 与 sentiment_score。系统同时保留 segment_results.csv 的片段级结构化字段和情绪分数；内置汇总仅按 source_file 与 source_row_index 回连，结构化字段按片段顺序去重后以中文分号拼接，情绪分数按片段字符数加权平均。

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
pandas	表格读取与汇总	读取 CSV、组织 DataFrame、写出 predictions_partial.csv 和 predictions_final.csv。
openpyxl	Excel 读写	通过 pandas 的 Excel 引擎读取 .xlsx/.xls；输出也可配置为 Excel。
transformers AutoTokenizer	文本编码与长度判断	计算 token 长度、生成 input_ids、attention_mask 和 token_type_ids。
transformers AutoModel + PyTorch	StructBERT 多标签推理	backbone 提取 CLS 向量，线性分类头输出 logits，sigmoid 后得到各情绪维度。
label_config.json	标签、中文名、阈值与权重	定义 7 个情绪维度的中文解释、0.5 阈值和 sentiment_score 的正负权重。
Qwen qwen-plus	结构化字段提取	读取用户填写的字段清单，逐行从原文提取明确出现的信息；缺失字段返回空字符串，并保留 structured_json 便于复核。

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. 准备表格	上传 .csv、.xlsx 或 .xls；默认每个文件中有 text 列。默认 input_glob=*.csv，处理 Excel 时把 INPUT_GLOB 改为 *.xlsx 或 *.xls。	源文件列表
2. 校验与读取	pandas/openpyxl 读取表格，检查 text 列存在且不全为空，记录 source_file 和 source_row_index。	文本行
3. 用户设置分段	网页可选择是否按标点切分，并填写每段最长字符数（50–20000）。开启时优先在 。！？；!?、换行处切开，再把相邻片段组合到字符上限；超长单句才硬切。	`segment_count / segment_max_chars`
4. 长文本分块	需要情绪时，StructBERT 还会在每个用户分段内部按 tokenizer 的 MAX_LENGTH=256 token 装箱，最终按原始行整合。	文本块

步骤	人话说明与复现动作	得到什么
5. 结构化提取	只有用户填写字段清单时，Qwen 才严格返回 JSON；公司名称、股票代码、产品名称等字段展开为独立列。	结构化字段列与 `structured_json`
6. 本地模型推理（按需）	只有情绪选项不是“无”时才运行 AutoTokenizer、StructBERT 和 sigmoid 情绪推理。	每块 7 维分数
7. 保留片段明细	每个分段单独写入 segment_results.csv，保存片段文本、序号、字符数、结构化字段和情绪分数，用户可以按自己的研究设计重新聚合。	`segment_results.csv`
8. 块级加权	需要情绪且一条原文分成多个块时，以每块去掉空白后的字符数作为权重，计算每个情绪维度的加权平均。	文本级情绪分数
9. 标签与综合分数	需要情绪时，最高分维度写入 top1_label；达到 thresholds 的维度写入 top_labels；按 weights 计算 sentiment_score。	`predictions_final.csv`
10. 保存与断点	每个源文件写入 per_file/<source>_predictions.csv，progress.json 保存最后完成行；全部完成后合并并写 run_summary/source_coverage/config_snapshot。	完整输出目录

五、公式、权重与融合规则

$$p_{i,k} = \frac{1}{1 + e^{-z_{i,k}}}$$

$$\bar{p}_{i,k} = \frac{\sum_j l_{i,j}\, p_{i,j,k}}{\sum_j l_{i,j}}$$

$$SentimentScore_i = \sum_k w_k\, \bar{p}_{i,k}$$

LaTeX 写法（可复制到论文编辑器）

```
p_{i,k}=\sigma(z_{i,k})=\frac{1}{1+e^{-z_{i,k}}}

\bar p_{i,k}=\frac{\sum_{j=1}^{m} l_{i,j}p_{i,j,k}}{\sum_{j=1}^{m}l_{i,j}}

\operatorname{SentimentScore}_i=\sum_{k=1}^{K}w_k\bar p_{i,k}
```

模型分数： $p_{i,k}=\text{sigmoid}(z_{i,k})=1/(1+e^{-z_{i,k}})$。z 是分类头对文本 i 在情绪维度 k 上输出的 logit，p 取值 0–1。

长文本聚合： $\bar{p}_{i,k}=\Sigma_{j=1}^{m} l_{i,j}p_{i,j,k}/\Sigma_{j=1}^{m}l_{i,j}$。l 是第 j 个文本块去掉首尾空白后的字符数，块越长，对原文结果影响越大。

综合情绪： $\text{SentimentScore}_i=\Sigma_k w_k\bar{p}_{i,k}$。当前权重为 optimism=+1、pessimism=-1、anxiety=-1、anger=-1、expectation=+0.5、neutral=0、uncertainty=-0.5。它们不是自动学习的权重，而是 label_config.json 中的预设。

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
`text`	原始文本
`segment_text`	切分后的片段原文
`segment_index`	片段在原始文本中的顺序，从 0 开始
`segment_chars`	片段字符数，用于情绪加权
`optimism`	乐观分数
`pessimism`	悲观分数
`anxiety`	焦虑分数
`anger`	愤怒分数
`expectation`	预期分数
`neutral`	中性分数
`uncertainty`	不确定性分数
`top1_label / top1_label_cn`	得分最高的英文标签及其中文名
`top_labels`	达到阈值的所有情绪中文标签，用 ; 连接
`sentiment_score`	按 label_config.json 权重加总的综合情绪分数
`source_file`	原始源文件名
`source_row_index`	原始文件中的行号索引
`prompt_fingerprint`	提示词和情绪配置的哈希，用于复现检查；不是模型分数
`segment_count / segment_max_chars`	原始文本行被拆成的段数及用户设置的字符上限

英文变量名	中文解释
split_by_punctuation	是否优先在标点和换行处切分
structured_json	Qwen 返回的结构化字段 JSON 快照
structure_status	结构化任务状态；未填写结构化字段时为 not_requested

七、结果文件与研究用途

结果文件	用途
outputs/segment_results.csv	每个分段一行的结构化与情绪明细，可自行重新聚合
outputs/predictions_final.csv	结构化字段与按需生成的情绪结果
outputs/structured_results.csv	仅结构化结果的清晰副本
outputs/emotion_results.csv	按原始文本行整合后的情绪结果；关闭情绪时不生成
outputs/predictions_partial.csv	运行中持续写入的中间结果
outputs/per_file/*_predictions.csv	每个源文件的独立结果
outputs/run_summary.json	行数、完成数、源文件覆盖情况
outputs/source_coverage.json / config_snapshot.json	源文件覆盖和本次参数快照

可以直接使用各维度连续分数，而不是只保留 top1_label；例如把 optimism、uncertainty 作为解释变量或控制变量，把 source_row_index 与 source_file 用于回溯原文。论文中应报告模型路径、标签配置、MAX_LENGTH、长文本加权方式和权重来源。

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。