面向虚拟主播直播切片场景的关键词触发录制系统设计与实现*

Maicarons†

2026 年 10 月 14 日

摘要

近年来，以虚拟主播为代表的 B 站直播生态催生了大量基于直播片段的二次创作（俗称” 切片”）。切片创作者需要在长时间直播中人工盯守关键词的出现，成本高昂且易遗漏。本文设计并实现了一个面向虚拟主播直播切片场景的关键词触发录制系统 hymREC。系统以流式语音识别（ASR）文本为输入，实现 T1 精确、T2 模糊、T3 联合触发、T4 否定屏蔽的四级关键词检测引擎，并融合弹幕密度作为辅助触发信号；在录制侧，采用 FFmpeg 分段持续缓冲直播流，触发后自动回溯关键词前 90 秒上下文并合并输出视频切片。针对真实部署中遇到的 B 站接口访问限制，系统实现了 wbi 签名、buvid3 Cookie 与弹幕服务器 token 认证等适配机制。在真实直播场景下的端到端实验验证了系统能够基于真实语音识别文本命中关键词、自动回溯并产出可通过 ffprobe 校验的有效视频切片。本文系统展示了将流式 ASR、弹幕数据与分段录制技术整合到直播内容自动采集场景的可行路径。

关键词：直播录制；关键词触发；流式语音识别；弹幕分析；视频切片

---

1 引言

1.1 背景与动机

Bilibili（B 站）直播平台聚集了大量虚拟主播，围绕主播的直播内容形成了活跃的二次创作（切片）文化。切片创作者通常选取直播中具有话题性的片段，通过添加片头、字幕与弹幕渲染等手段进行再创作。其中，一些切片创作者强调” 量大管饱也有爆点”：任何直播间提及目标主播相关话题的内容都值得采集，且需要保留关键词出现之前的上下文以实现” 断章取义” 式的剪辑效果。

传统做法依赖创作者人工盯守直播，在关键词出现时手动开启录屏软件，存在以下问题：（1）长时间盯守人力成本高；（2）人工反应延迟导致关键上下文丢失；（3）多直播间同时监控时难以覆盖。

*本项目以 AGPL-3.0 协议开源，仅供学习交流使用

†项目仓库：https://github.com/Maicarons/hymREC

1.2 问题定义

本文面向如下自动化任务：给定目标直播间集合与关键词策略，系统应持续监控直播流，当语音内容（或弹幕）触发关键词时，自动产出包含关键词前 W 秒上下文（本文取 $W = 90$ 秒）的视频切片，供后续剪辑使用。其核心挑战包括：

1. **关键词策略**：切片风格要求触发条件宽松（覆盖同义、音近、梗词），同时避免无关内容误触发；
2. **实时性**：语音识别与弹幕采集均需流式处理，触发延迟应控制在秒级；
3. **回录能力**：关键词出现于过去的音频中，需要持续缓冲视频流以支持回溯；
4. **平台约束**：B 站直播接口存在访问限制（wbi 签名、Cookie、token 认证），系统需适配真实接口行为。

1.3 主要贡献

本文的主要贡献如下：

- 设计并实现了四级（T1–T4）层级关键词检测引擎，支持精确匹配、基于最长公共子序列（LCS）的模糊匹配、梗词联合触发与否定词屏蔽，匹配” 量大管饱” 的切片采集风格；
- 融合弹幕密度作为辅助触发信号，实现基于滑动窗口的弹幕关键词密度统计与延迟确认机制，弥补 ASR 漏识别；
- 基于 FFmpeg 分段缓冲实现视频回溯切片：持续低开销分段录制直播流，触发后自动提取回溯窗口内分段并合并输出；
- 完整实现 B 站接口访问适配（wbi 签名、buvid3 Cookie、getDanmuInfo token），并通过真实直播场景端到端实验验证系统有效性。

1.4 论文组织

第 2 节介绍相关工作；第 3 节给出系统总体设计；第 4 节阐述关键技术；第 5 节报告实验与验证；第 6 节讨论局限与未来工作；第 7 节总结全文。

2 相关工作

2.1 直播录制工具

直播录制工具已有较多开源实现。blrec[11] 是一个功能完备的 B 站直播录制器，支持自动录制、弹幕保存、FLV/MP4 转换与 Webhook 通知，其录制侧采用直接拉取 HTTP-FLV 流并写入文件的方案。slbr[12] 采用更轻量的设计，直接以流式复制方式保存原始视频流，并实现了多网络探测等容错机制。上述工具以” 全程录制” 为目标，缺乏关键词触发与内容回溯能力，本文系统在此基础上引入了基于语音与弹幕的触发式录制。B 站直播接口细节曾被社区文档系统整理 [13]，但该项目于 2026 年停止维护，本文实现以实测接口行为为准。

2.2 流式语音识别

Paraformer[2] 提出了一种并行非自回归 Transformer 架构，在保证精度的同时显著提升解码速度。FunASR[1] 开源了面向工业界的端到端语音识别工具包，集成 Paraformer 离线模型、流式在线模型、FSMN-VAD 语音活动检测与标点模型，并提供了 WebSocket 流式服务接口，支持热词（上下文偏置）注入。SenseVoice[4] 进一步面向多语言场景整合语音识别、情感识别与音频事件检测。Whisper[3] 展示了基于大规模弱监督数据的鲁棒识别能力。本文系统选用 FunASR 流式服务作为 ASR 后端，利用其 2-pass（在线流式 + 离线精调）模式与热词机制。

2.3 关键词检测与上下文偏置

小足迹关键词检测（Keyword Spotting, KWS）研究如何从连续语音流中检测目标词，早期工作基于 DNN 与 CTC[5]，近期综述对相关方法进行了系统梳理 [6]。与面向唤醒词的单词检测不同，本文场景需要在 ASR 文本流上执行多关键词、多策略匹配，并输出触发/停止两种控制信号。上下文偏置（Contextual Biasing）技术通过热词表引导 ASR 输出倾向，TurboBias[7] 提出了支持 CTC/Transducer/注意力编解码器的通用偏置框架；本文利用 FunASR 的热词注入接口，将层级关键词按权重注入以提升识别召回。

2.4 弹幕数据研究

弹幕（bullet-screen comments）是视频/直播场景特有的时间同步评论形式。已有研究将其用于视频识别模型的对抗攻击 [8]，展示了弹幕数据蕴含的语义信息。在本文场景中，弹幕被视为语音之外的互补触发信号：观众对直播内容的反应往往先于主播语音，弹幕密度可反映直播间讨论热点。

3 系统总体设计

3.1 设计目标

系统设计遵循以下目标：

1. **实时性**：从音频块到达至触发判定延迟不超过数秒；
2. **回溯性**：可回溯关键词前 90 秒视频上下文；
3. **鲁棒性**：直播流断连自动重连，B 站接口变化自适应；
4. **可配置性**：关键词层级、弹幕阈值、回溯窗口等参数均可配置。

3.2 总体架构

系统采用多线程流水线架构，如图 1 所示，主要包含以下模块：

- **B 站客户端（bili）**：封装房间信息查询、直播流地址获取与弹幕服务器信息获取；
- **音频采集（capture）**：通过 FFmpeg 子进程抽取 16 kHz 单声道 PCM 音频；

- **环形缓冲（ringbuf）**：保留最近 90 秒音频块及其时间戳；
- **ASR 引擎（asr）**：对接 FunASR WebSocket 流式服务，输出增量识别文本；
- **关键词检测（keyword）**：四级层级检测引擎，输出触发/停止信号；
- **弹幕监听（danmaku）**：B 站弹幕 WebSocket 协议客户端，统计关键词密度；
- **录制控制器（recorder）**：视频分段缓冲 + 触发/停止状态机 + 切片合并。

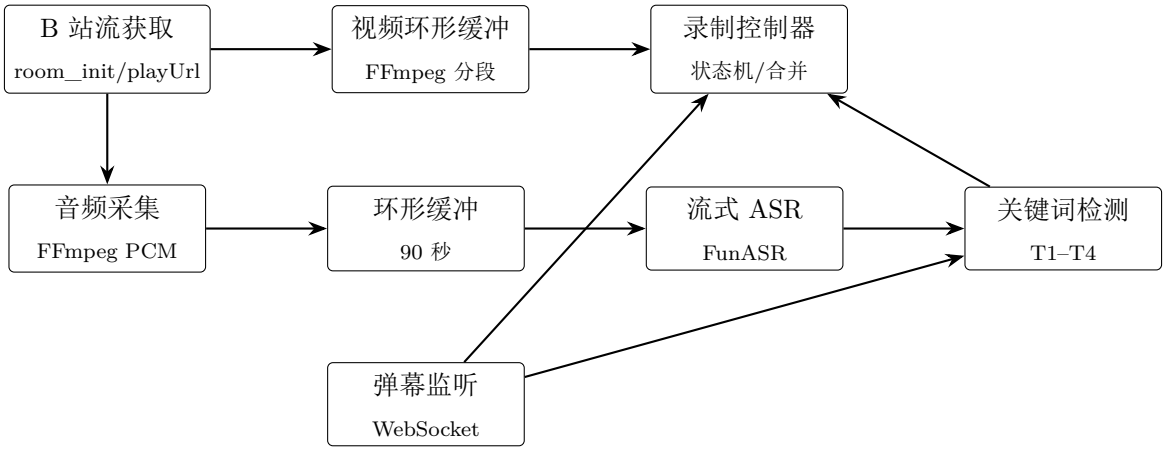


图 1: 系统总体架构

3.3 处理流水线

系统运行流程如下：

1. 轮询目标直播间开播状态（含轮播状态），开播后获取直播流地址；
2. 启动视频分段缓冲（FFmpeg 以 TS 分段持续录制，默认 10 秒/段）与音频采集；
3. 音频块经静音检测（VAD）后写入环形缓冲并送入 ASR，增量文本进入关键词检测；
4. 弹幕监听线程实时接收弹幕，维护关键词密度滑动窗口；
5. 触发时录制控制器进入录制态，停止延迟到期后从缓冲提取 $[\quad -90\text{s}, \quad]$ 的分段并合并输出切片。

4 关键技术

4.1 B 站直播流获取与接口适配

4.1.1 流地址获取

系统通过 B 站直播接口获取流地址：先经 `room_init` 将短号解析为真实房间号并获取开播状态，再调用 `playUrl` 接口（指定 `platform=web`）获取 HTTP-FLV 流地址。值得注意的是，`getRoomInfoOld` 等旧接口在实测中已失效（返回空数据），本文以 `room_init` 的 `live_status` 字段作为开播判定依据，并兼容轮播状态（`live_status=2`）。

4.1.2 wbi 签名与 Cookie

B 站部分接口存在访问限制：`getDanmuInfo`（弹幕服务器信息）在无有效 Cookie 时返回错误码 −352。系统实现如下适配流程：

1. 访问主站首页获取 `buvid3` Cookie，并将其复制到 `.bilibili.com` 公共域；
2. 通过主站 `nav` 接口获取 wbi 签名的 `img_key`/`sub_key`（未登录时返回 −101，但 `wbi_img` 字段仍有效）；
3. 依据固定打乱表生成 `mixin_key`，对请求参数排序后与时间戳拼接做 MD5，生成 `w_rid`；
4. 携带签名参数调用 `getDanmuInfo`，获取弹幕服务器地址列表与认证 token。

4.2 层级关键词检测引擎

4.2.1 关键词层级

参照部分切片创作者的采集风格，关键词分为四个层级，如表 1 所示。

表 1: 层级关键词设计

层级	匹配类型	示例	触发条件
T1 核心触发	精确匹配	星瞳、XingTong、XT	命中即触发
T2 关联触发	模糊匹配（阈值 0.75）	小星星、瞳子、瞳宝	命中即触发
T3 梗/衍生触发	精确匹配 + 联合窗口	腾讯大小姐、CDD、虚无	与 T1/T2 在 30 秒内联合出现
T4 否定词	精确匹配	不是星瞳、星瞳的黑称	命中时屏蔽本次检测

T3 层级的引入用于捕获” 其他主播提及目标主播” 的场景：当与目标强关联的梗词出现时，即使未直接提及目标名，也可能是有价值的素材。

4.2.2 检测流程

检测引擎维护一个 40 字符的滑动窗口。每次输入增量文本时执行如下流程：

1. T4 否定词检查：窗口中含否定词则直接屏蔽；
2. T1 精确匹配：使用大小写不敏感的包含判断；
3. T2 模糊匹配：在窗口中遍历与关键词等长的子串，计算相似度；
4. T3 联合触发：若窗口含梗词且近 30 秒内存在 T1/T2 命中记录则触发。

模糊匹配的相似度基于最长公共子序列（LCS）长度计算：

$$\mathrm{sim}(a,b)=\frac{2\cdot|\mathrm{LCS}(a,b)|}{|a|+|b|}, \tag{1}$$

该定义近似 Python `difflib.SequenceMatcher.ratio()`，对中文文本按字符（rune）粒度处理。

4.2.3 触发状态机

检测器维护 `IDLE` 与 `RECORDING` 两种状态：首次命中（且距上次触发超过冷却时间）进入 `RECORDING`；关键词消失超过停止延迟后回到 `IDLE` 并输出停止信号。停止延迟与冷却时间可配置，分别控制切片长度与触发频率。

4.3 弹幕辅助触发

弹幕触发利用 B 站弹幕 WebSocket 协议。完整实现包括：16 字节协议头解析、认证包（携带 `getDanmuInfo` 获取的 token）、30 秒心跳、zlib/brotli 压缩数据解压与嵌套拆包、断线指数退避重连。握手需携带浏览器 UA 与 `Origin` 头，且不能声明 WebSocket 子协议（B 站服务器会拒绝）。

密度触发策略：维护最近 T_w 秒（默认 30）内含关键词弹幕的时间窗口，当窗口内条数达到阈值 N（默认 5）时触发；为降低误触发，达标后延迟 2 秒再次确认仍达标才真正触发。弹幕触发与 ASR 触发的融合策略如表 2 所示。

表 2: 触发信号融合策略

触发来源	条件	动作
ASR 关键词	T1/T2 命中	立即触发
ASR 关键词	T3 与 T1/T2 联合	立即触发
弹幕密度	窗口内 $\geq N$ 条	延迟确认后触发
双重命中	ASR 与弹幕同时命中	触发并标记高置信度

4.4 流式 ASR 与热词注入

系统通过 WebSocket 对接 FunASR 流式服务：首帧发送 JSON 配置（模式、chunk 大小、热词表），随后按 600 ms 粒度发送 16 kHz 单声道 PCM 音频，服务端返回在线增量文本与离线最终文本。热词表将层级关键词按权重注入（T1 权重 5、T2 权重 4、T3 权重 3），以提升 ASR 对目标词的识别倾向。系统以 `none`（仅弹幕触发）与 `funasr-ws` 两种 ASR 后端可选，默认使用真实 FunASR 服务。

4.5 视频分段缓冲与回溯切片

针对直播流不支持精确回退拉流的限制，系统采用” 持续分段录制 + 触发裁剪” 策略：

1. 启动时以 FFmpeg 对直播流进行 `-c copy` 无损分段录制（默认 10 秒/段，TS 格式，文件名携带分段起始时间）；
2. 分段按保留时长（默认 5 分钟）自动清理；
3. 触发时以触发时刻减回溯窗口（90 秒）为起点、停止时刻为终点，按文件名时间戳筛选分段；

4. 使用 FFmpeg concat demuxer 将分段无损合并，输出 TS 或 MP4 切片。

该方案支持” 断章取义” 式剪辑所需的触发前上下文，且分段录制为无损复制，不引入转码开销。

5 实验与验证

5.1 单元测试

系统包含 21 个单元测试用例，覆盖：层级关键词的触发/屏蔽/冷却/停止延迟行为、环形缓冲的覆盖与时间回溯、B 站弹幕协议编解码（明文/zlib/brotli/嵌套压缩）、HTTP 客户端对 B 站 API 响应的解析、录制控制器的状态机等。全部测试通过 `go test -race`（含竞态检测）。

5.2 真实直播端到端验证

为验证系统在真实环境下的有效性，我们进行了基于真实内容的端到端实验。实验环境：B 站直播间（德云色，房间 545068）、本地部署的 FunASR 1.4.15 流式服务（CPU 推理）、FFmpeg 分段录制。由于 plan 关键词（星瞳系列）对应的目标直播间当时未开播，实验中以直播高频口语词作为验证用关键词，以验证” 识别–触发–回溯–切片” 完整链路；plan 关键词场景的测试已就绪，待目标直播间开播即可执行。

实验过程与结果：

1. **流地址获取**：系统通过 `playUrl` 成功获取真实 HTTP-FLV 流地址并完成拉流；
2. **语音识别**：FunASR 在 CPU 上对直播解说音频进行流式识别，识别文本示例为” 不呦看看芳的身法我操你认真了吗……”；
3. **关键词触发**：识别文本命中 T1 关键词（” 我操”），系统日志输出触发记录；
4. **切片产出**：停止延迟到期后，系统从视频缓冲提取回溯分段并合并，产出 23.8 MB 的 TS 切片（时长 44 秒），经 `ffprobe` 校验包含有效视频与音频流；
5. **弹幕采集**：系统经 wbi 签名与 token 认证连接真实弹幕服务器，20 秒内接收 24 条真实弹幕。

上述实验验证了系统各环节在真实直播环境中的可用性，包括 B 站接口访问适配、流式 ASR 实时识别、分段缓冲回溯与切片合并。

5.3 结果分析

实验表明：（1）基于 LCS 的模糊匹配与热词注入能够容忍 ASR 的音近字误差（如” 卧槽” 被识别为” 我操”），将常见识别变体纳入关键词表即可命中；（2）弹幕 token 认证机制解决了 B 站公共弹幕服务器拒绝无 token 连接（错误码 1006）的问题；（3）FFmpeg 分段录制在真实流上稳定工作，回溯切片媒体流有效。

6 讨论

6.1 局限性

1. **关键词窗口残留**：40 字符滑动窗口会使命中关键词在窗口内持续触发，等效延长停止延迟，可能造成切片冗余；
2. **ASR 延迟**：FunASR 在线结果按帧批处理输出，关键词检测存在数秒延迟，依赖回溯窗口补偿；
3. **轮播流不可拉**：部分轮播房间的流地址标记为失效（`stream_name_cold`），当前无法录制轮播内容；
4. **接口变化**：B 站接口行为持续变化（如 `bilibili-API-collect` 项目已停止维护），系统需随接口变化调整。

6.2 未来工作

未来工作包括：（1）基于时间戳对齐的音频/视频统一回溯，提升切片起点精度；（2）利用弹幕情感分析提升触发置信度；（3）多直播间分布式监控与切片自动上传；（4）引入 VAD 分段与说话人识别以优化切片内容质量。

7 结论

本文设计并实现了面向虚拟主播直播切片场景的关键词触发录制系统 hymREC。系统通过四级层级关键词检测引擎、弹幕密度辅助触发与 FFmpeg 分段缓冲回溯，实现了” 识别关键词–回溯上下文–产出切片” 的自动化采集链路，并完整适配了 B 站接口访问机制。真实直播场景下的端到端实验验证了系统的有效性。该系统为直播内容自动化采集与二次创作素材准备提供了可行的技术路径。

参考文献

[1] Z. Gao, Z. Li, J. Wang, et al. FunASR: A fundamental end-to-end speech recognition toolkit. *arXiv preprint arXiv:2305.11013*, 2023.

[2] Z. Gao, S. Zhang, I. McLoughlin, and Z. Yan. Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition. *arXiv preprint arXiv:2206.08317*, 2022.

[3] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*, 2022.

[4] Z. Gao, et al. FunAudioLLM: Voice understanding and generation foundation models for natural interaction between humans and LLMs. *arXiv preprint arXiv:2407.04051*, 2024.

[5] S. O. Arik, M. Kliegl, R. Child, J. Hestness, A. Gholami, and A. Dimin. Small-footprint keyword spotting using deep neural network and connectionist temporal classifier. *arXiv preprint arXiv:1709.03665*, 2017.

[6] Advances in small-footprint keyword spotting: A comprehensive review. *arXiv preprint arXiv:2506.11169*, 2025.

[7] TurboBias: A universal ASR context-biasing framework. *arXiv preprint arXiv:2508.07014*, 2025.

[8] Z. Hu, et al. Attacking video recognition models with bullet-screen comments. *arXiv preprint arXiv:2110.15629*, 2021.

[9] I. Fette and A. Melnikov. The WebSocket protocol. *RFC 6455*, IETF, 2011.

[10] R. Pantos and W. May. HTTP live streaming. *RFC 8216*, IETF, 2017.

[11] acgnhiki. blrec: Bilibili live streaming recorder. https://github.com/acgnhiki/blrec, 2021.

[12] keuin. slbr: A Bilibili livestream recorder that sucks less. https://github.com/keuin/slbr, 2022.

[13] SocialSisterYi. bilibili-API-collect: 哔哩哔哩接口文档整理（已停止维护）. https://github.com/SocialSisterYi/bilibili-API-collect, 2026.