**文案全链路监控优化**

1. **背景**

在当前电商各平台中，存在很多用户难以理解的文案；这类文案通常散落在各个不同的业务场景、有着不同的展现方式；通常这类文案会对用户使用我们电商平台造成较多的干扰，影响用户体验，造成用户进线；

当前用户体验、用户费力度在整个平台中都是非常重要的业务指标。而文案又是实实在在影响着用户体验，起着指导用户理解产品意图，指引用户操作的关键作用。

所以建设一个可以监控、度量当前所有平台文案质量的手段是必要的。

**常见问题如下**

* 提示文案难以理解
* 查询行业规则RPC异常
* 创建失败，请稍后重试(code=20000000)
* 客服在会话7360628642608694547发送诱导脱离平台交易信息（如第三方链接、联系方式、货到付款等）
* 店铺发布的商品缺少必要信息[奖惩业务文案数据](https://bytedance.larkoffice.com/docx/CBsidOFsGoEE3OxuV6ecpStjnMe)
* 文案中包含错别字
* 该登陆账号无im权限
* 文案冗余用户难以理解



* 文案中英文标签混用
* 商家出售商品或提供服务时盗用他人图片.
* 专业名词影响用户体验
* 24H支付揽收率

**文案全链路监控优化的目的就是为了监控、度量各个平台的文案质量，通过大模型给出文案不同维度的分数，帮助电商各平台在文案规范、文案质量上都成为行业标杆。**

2. **预期收益**

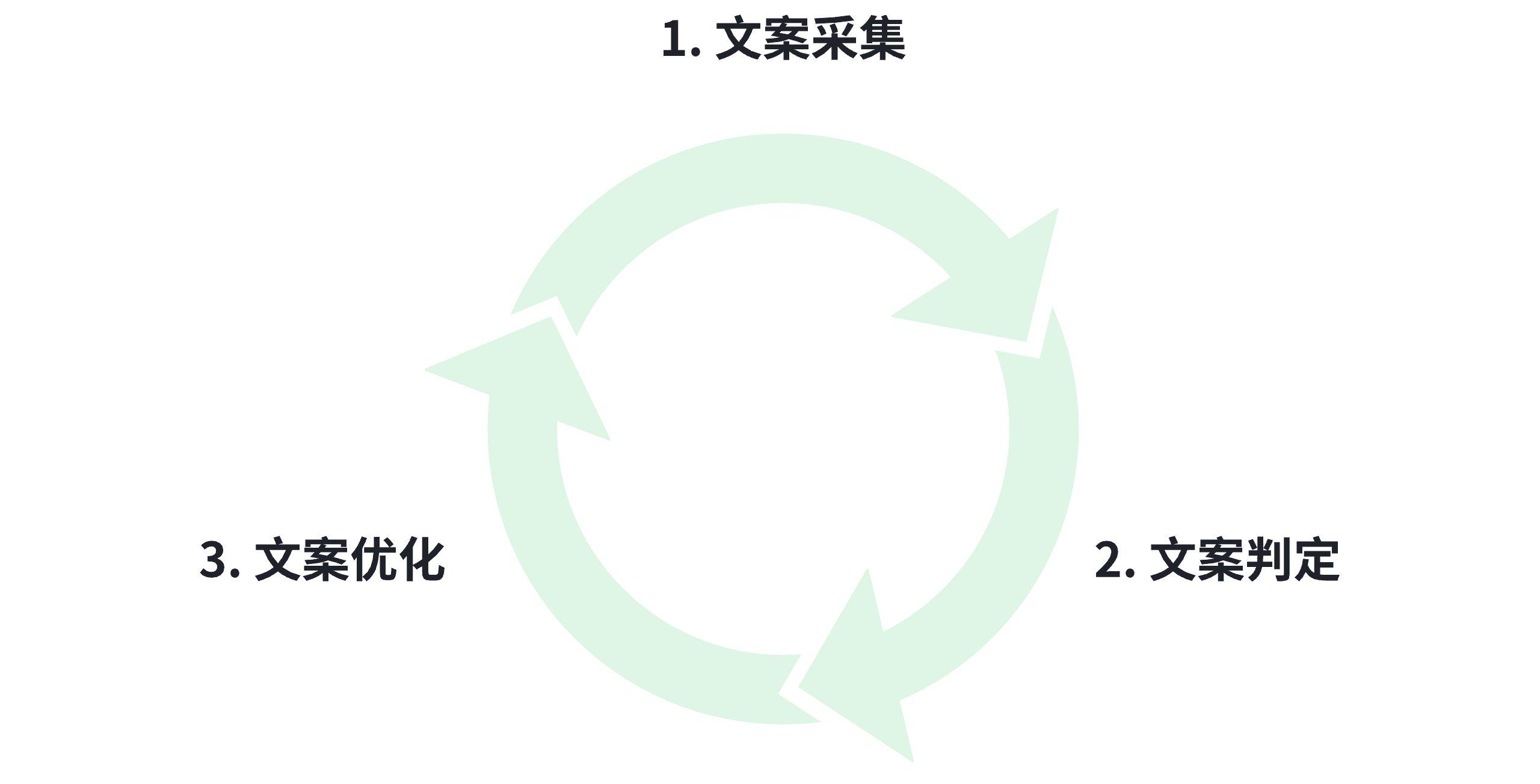
整体收益：从【采集 → 判定 → 优化 → 投放】全链路监控产品文案，低成本、可信、可持续优化产品文案体验

详细收益：

|  |
| --- |
| 收益点：   * 业务收益 * 通过文案质量监控，推进各业务方完成低质文案优化，降低相关 CPO（文案在 CPO 的表现上场景较多：指标定义咨询、不清楚怎么操作、不了解规则等） * 帮助 xx 业务提升用户理解度 * 奖惩业务 30% -> 60% * 规范平台文案规则；将错字、中英文混用降至 0，提升品牌形象 * 抖店、即时零售等平台文案分数 提升至 xx * 支撑 xx 业务快速接入文案反馈，帮助提高用户理解度 * 大模型文案优化采纳率达到 xx * 技术收益 * 支持各平台快速完成文案分数接入 * 支持文案用户反馈快速接入，支持文案富文本展示，提升研发效率 xx * 大模型能力中台化，提供快速 few-shots 支持，提供模型对比能力 * 文案数据从天提升到小时级 * 解决之前正则识别错误的文案 * 丰富文案数据链路 |

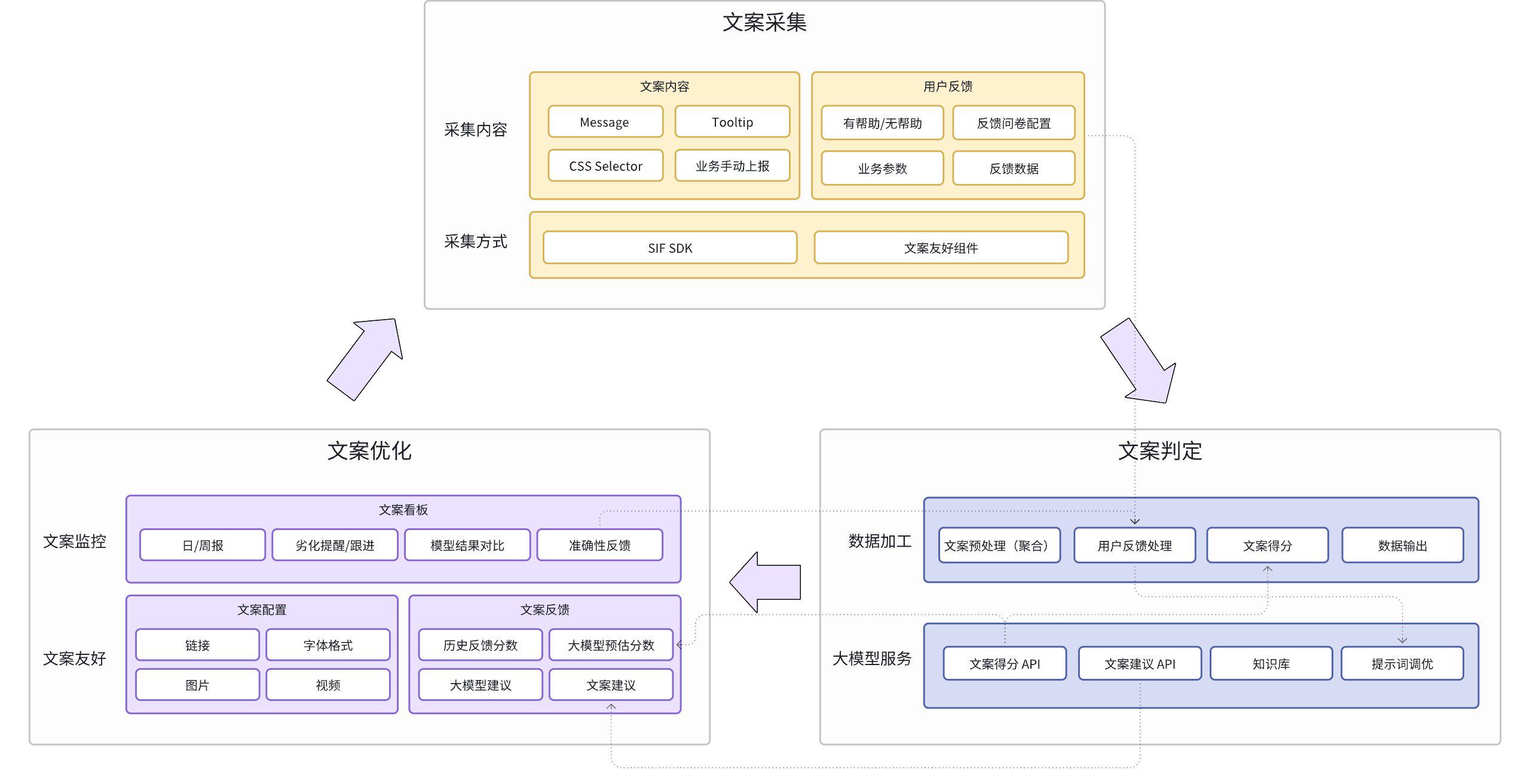
|  |  |  |
| --- | --- | --- |
| 阶段 | 收益点 | 图示 |
| 采集 | **自动采集**用户端上看到的文案 | |  |  | | --- | --- | |  |  | |
| 使用现有的文案组件，**可极低成本收集用户反馈** | |  |  | | --- | --- | |  |  | |
| 判定 | 通过设立文案**评分标准**，通过**大模型能力**对文案评分，保证**评分结果可信**  [大模型评分标准](https://bytedance.larkoffice.com/docx/SUkEdm1zFoWBs2xWUJpcSIBYnGc) |  |
| 建立大模型**评分优化机制**，保证大模型评分置信并可持续迭代（评分标准更新、用户反馈） |  |
| 优化 | 产出**每日文案评分看板**，定期跟进优化，支持**劣化提醒**，持续优化文案 | |  |  | | --- | --- | |  |  | |
| 提供富文本能力，支持在文案中**插入链接、图片、视频**，提高可读性、可理解性 |  |
| 提供**后台管理能力**，产品运营可直接在运营后台配置文案。降低文案更新、修改成本 |  |
| 结合评分标准，利用大模型能力提供 **AI 文案优化建议** |  |

3. **整体方案**

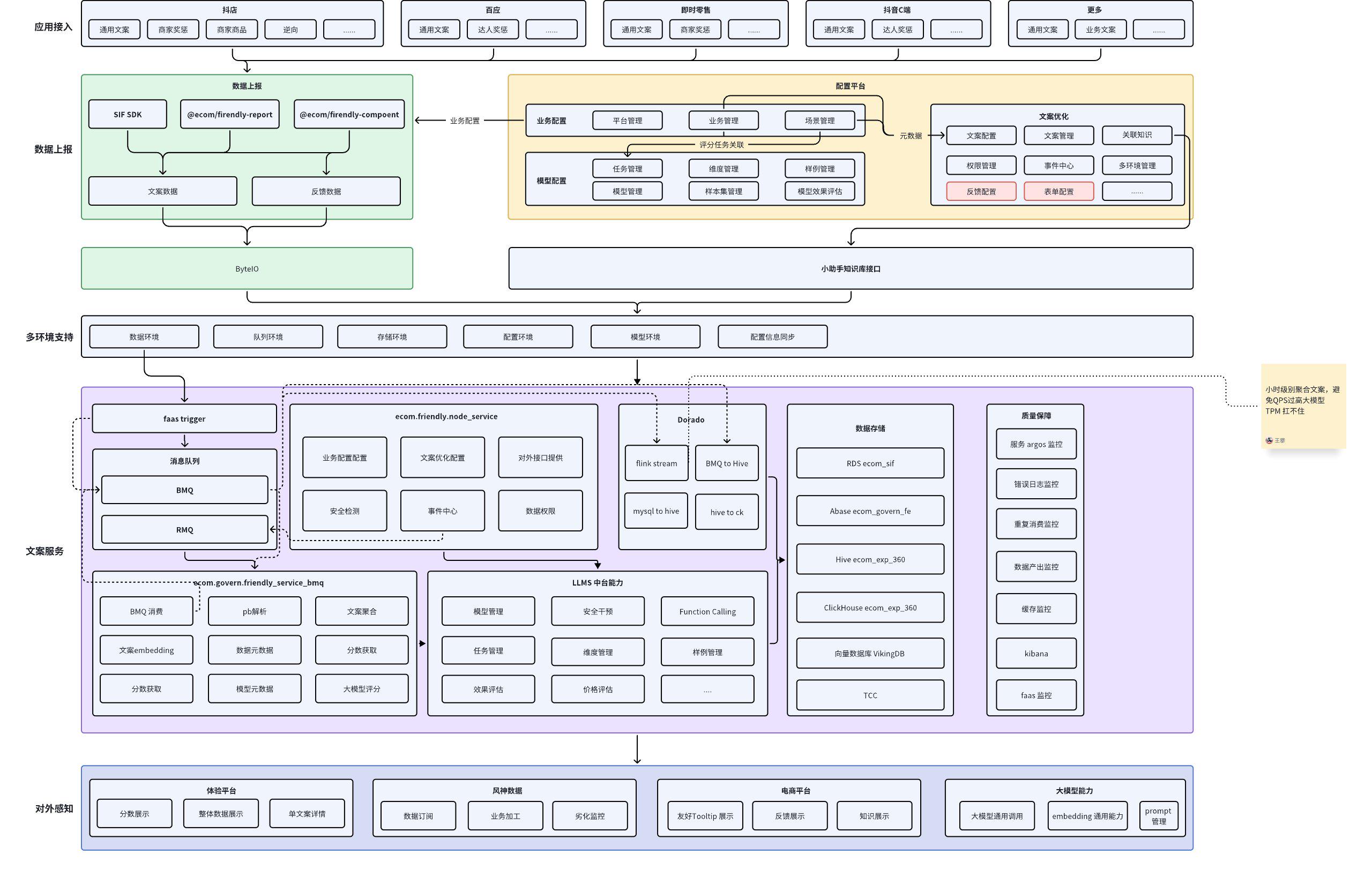


1. **文案采集**：采集相关产品上展示的所有文案及用户反馈
2. **文案判定**：通过策略、大模型、用户反馈判断采集上来的文案的可读性和可理解性
3. **文案优化**：对于可读性、可理解性低的文案，从优化建议、展示方式等方面对文案做优化。优化之后再投放，重新走到采集过程，收集文案及用户反馈

**产品流程图**



**技术架构**



4. **研发资产**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 资产名称 | 资产地址 | 资产明细 | 资产说明 | 页面地址 |
| 仓库 | https://code.byted.org/ecom/friendly-information | [ecop-friendly-information](https://code.byted.org/ecom/friendly-information/tree/master/apps/ecop-friendly-information) | 运营后台相关配置页面 | [文案管理地址](https://ecop.bytedance.net/friendly-information/manage?cjSiteCode=St12209160001002&cjDomainCode=9&__domainId=9&btm_ppre=a0.b0.c0.d0&btm_pre=a41971.b835895.c0.d0&btm_show_id=ac66caab-431b-4ea1-a0f8-7ff507ca461a)  [文案配置地址](https://ecop.bytedance.net/friendly-information?cjSiteCode=St12209160001002&cjDomainCode=9&__domainId=9) |
| [friendly-service](https://code.byted.org/ecom/friendly-information/tree/master/apps/friendly-service) | 文案接口服务：目前主要包含了文案配置后台接口，文案、反馈表单下发接口；大模型相关配置页面 |  |
| [friendly-service-bmq](https://code.byted.org/ecom/friendly-information/tree/master/apps/friendly-service-bmq) | 文案 BMQ 消息处理服务，主要处理 BMQ 过来的消息 |  |
| [friendly-component](https://code.byted.org/ecom/friendly-information/tree/master/packages/friendly-component) | 文案组件：包含移动端 & PC 端 | [组件文档](https://module-docs.bytedance.net/package/@ecom/friendly-component/1.0.1/web/feedback-tooltip.html) |
| [friendly-core](https://code.byted.org/ecom/friendly-information/tree/master/packages/friendly-core) | 文案数据逻辑 |  |
| [friendly-report](https://code.byted.org/ecom/friendly-information/tree/master/packages/friendly-report) | 文案上报 |  |
| [gulux-plugin-coze](https://code.byted.org/ecom/friendly-information/tree/master/packages/gulux-plugin-coze) | Coze 插件 |  |
| Dorado | [Dorado 项目地址](https://data.bytedance.net/dorado/development/node/112817662?project=cn_6437) |  | 离线、流式数据处理 |  |
| FaaS | [faas trigger 地址](https://cloud.bytedance.net/faas/function/240yj502/cluster/detail?cluster=faas-cn-north&region=cn-north&x-resource-account=public) |  | Faas bmq trigger |  |
| TCE | [文案服务地址](https://cloud.bytedance.net/tce/services/6243178?cluster-type=cluster&module=cluster&page=1&page_size=10&x-resource-account=public) |  | 文案相关服务 |  |
| Goofy Deploy | [service\_bmq](https://deploy.bytedance.net/app/84702/deploy_unit_list?deployUnitId=131922) |  | Bmq 服务 |  |
| [配置后台](https://deploy.bytedance.net/app/63987/deploy_unit_list?deployUnitId=106731) |  | 文案配置后台 |  |
| BMQ | [BMQ 地址](https://cloud.bytedance.net/message-queue/topics?region=online&x-resource-account=public&activeTab=my) | [sif\_text\_topic](https://cloud.bytedance.net/message-queue/topics/1031946/detail?region=online) | 文案聚合 topic |  |
| [sif\_text\_score\_origin](https://cloud.bytedance.net/message-queue/topics/1040824/detail?region=online) | 文案分数原始到 hive topic |  |
| [sif\_text\_flint\_to\_hive\_topic](https://cloud.bytedance.net/message-queue/topics/1032450/detail?region=online) | 文案聚合数据到 hive topic |  |
| [sif\_friendly\_text\_feedback\_detail](https://cloud.bytedance.net/message-queue/topics/1036729/detail?region=online) | 文案反馈详情到 hive topic |  |
| [sif\_friendly\_text\_feedback\_action](https://cloud.bytedance.net/message-queue/topics/1036728/detail?region=online) | 文案动作到 hive topic |  |
| TCC | [TCC 配置中心](https://cloud.bytedance.net/tcc/namespace/ecom.sif.sdk_config?by_key=false&condition=name&dir_path=%2Fdefault&env=prod&filter_no_tag=false&keyword=&order=&pn=1&region=CN&rn=10&scope=all&x-resource-account=public) |  |  |  |
| RDS | [ecom\_sif RDS](https://cloud.bytedance.net/rds/detail/db/cn/ecom_sif/overview?x-resource-account=public) |  | 数据库地址 |  |
| Abase | [ecom abase 地址](https://cloud.bytedance.net/abase/abase2/ecom_govern_fe/4895/common7?%3F%3FactiveTab=tables&activeTab=authManage&exclusive=1&region=online&x-resource-account=public) |  | Cache |  |
| VikingDB | [babi 地址，需要从这里的电商账号下进入](https://babi.bytedance.net/finance/basic/volcManage/?fullscreen=true) |  | 选择 ecom\_sif 点击跳转火山 |  |
| GPT 中台 | [GPT 中台](https://gpt.bytedance.net/gpt_openapi/model?scene_keyword=ecom_text_score) |  |  |  |
| 火山方舟 | [babi 地址，需要从这里的电商账号下进入](https://babi.bytedance.net/finance/basic/volcManage/?fullscreen=true) |  |  |  |
| Lander 项目地址 | [lander](https://lander.bytedance.net/app/dashboard/project?id=10206) |  |  |  |

5. **详细方案**

5.1 **文案采集**

5.1.1 **自动采集方案**

|  |
| --- |
| 基于 SIF SDK 进行上报：  [体验平台事件采集（SIF SDK）One Page](https://bytedance.larkoffice.com/docx/WTYld6IedoEkQ9xJlOdcouHbnBh)  [SIF SDK 技术设计](https://bytedance.larkoffice.com/docx/IHFBdoriWouRK6xep5hcoPvxnxh) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 文案类型 | 组件名称 | 组件使用场景 | 采集方式 | 当前进展 | 示例 |
| 通用组件文案 | Message | 通常是一个操作之后的反馈，全页面可见 | 自动采集 | 已采集 |  |
| Popover / Tooltip | 通常是对一个定义、指标、文案的解释 | 接入文案友好组件可自动采集 |  |
| Modal | 通常是一段解释文案或者可以继续操作的提示文案 | 暂未采集 |  |
| 业务文案 | 无固定组件 | 通常是业务自定义的文案 | 手动上报（后续体验平台配置） | 奖惩业务文案 |  |

5.1.2 **业务手动上报**

|  |
| --- |
| 提供 [@ecom/friendly-report](https://bnpm.bytedance.net/package/@ecom/friendly-report) 进行手动上报  提供业务组件 [@ecom/friendly-component](https://bnpm.bytedance.net/package/@ecom/friendly-component) 进行自动上报 |

|  |  |  |  |
| --- | --- | --- | --- |
| 上报方式 | 上报方法 | 使用方式 | 字段解释 |
| 手动上报 | useSifTextReport | |  | | --- | | TypeScript import { useSifTextReport } from '@ecom/friendly-report'; useSifTextReport({  sif\_bid： 1,  business\_id: 10,  content: show\_violation\_reason,  scene\_id: sifRegulationViolationScene.violationReason,  extra: {  scene\_cn: '违规管理：违规处置类型',  rule\_id,  rule\_name: show\_violation\_reason,  ticket\_id  } }) | | * sif\_bid：sif app id * business\_id：业务ID * scene\_id：场景ID * cotent：文案内容 * extra：业务参数 |
| 组件上报 | FriendlyContent | |  | | --- | | TypeScript import { PageConfig, FriendlyContent } from '@ecom/friendly-component'; const Content = () => {  return (  <div>  <FriendlyContent textId="mxXQFyaafIqGmBlvOJwhw" />  </div>  ); };  export default function Page() {  return (  <PageConfig platformId={4272} pid="/ffa/eco/experience-score">  <Content />  </PageConfig>  ); } | | [文案友好组件](https://module-docs.bytedance.net/package/@ecom/friendly-component/1.0.0/web/friendly-tooltip.html) |
| FriendlyTooltip | |  | | --- | | TypeScript import { PageConfig } from '@ecom/friendly-component'; import { FriendlyTooltip } from '@ecom/friendly-component/web';  const MyTip = () => {  return (  <div>  <FriendlyTooltip textId="mxXQFyaafIqGmBlvOJwhw">24小时支付-揽收率</FriendlyTooltip>  </div>  ); };  export default function Page() {  return (  <PageConfig platformId={4272} pid="/ffa/eco/experience-score">  <MyTip />  </PageConfig>  ); } | |

5.1.3 **反馈数据上报**

|  |
| --- |
| 提供业务组件 [@ecom/friendly-component](https://bnpm.bytedance.net/package/@ecom/friendly-component) ，支持在 PC、H5 中接入 |

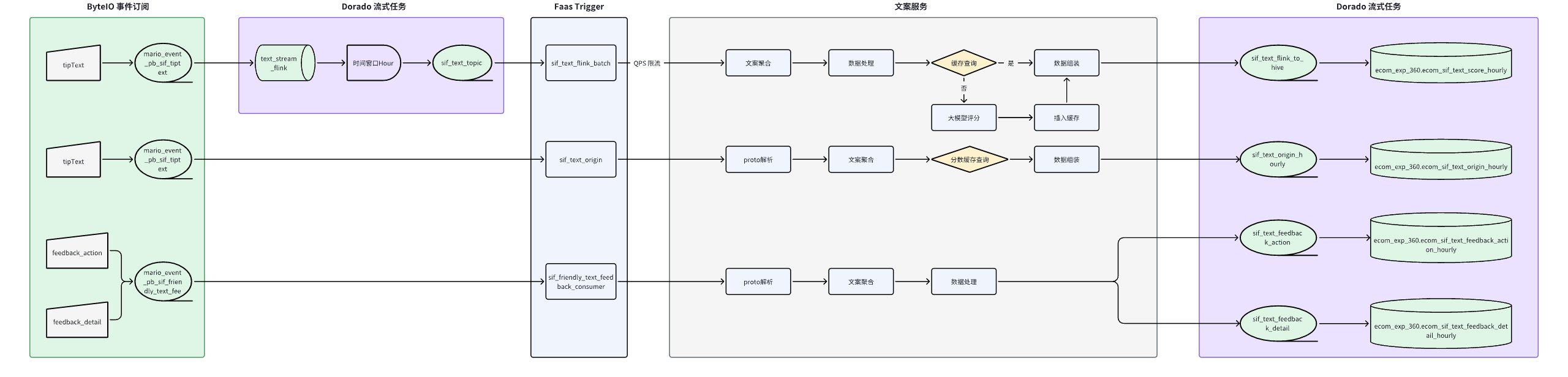
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 反馈组件 |  | 样例 | 接入方式 | 字段解释 |
| FeedbackEntry | PC |  | |  | | --- | | TypeScript import { FeedbackEntry } from '@ecom/friendly-component/web'; <FeedbackEntry  style={{ marginTop: 12 }}  teaData={{  business\_id: '15',  scene\_id: 'couCpwTuMsd\_W9qIXDjpN',  platform: '1128',  sif\_bid: 1128,  extra: {  governance\_id,  secUserId  },  content: '能否清晰地明白违规内容与原因？'  }}  helpfulText="看得懂"  unhelpfulText="看不懂"  unhelpfulId="detail\_collect\_unhelpful"  helpfulId="detail\_collect\_helpful"  colletHelpfulReason /> | | * teaData 反馈上报参数 * sif\_bid：sif app id * business\_id：业务ID * scene\_id：场景ID * 业务自定义参数 * content：对应文案 * helpfulText 有帮助文案 * unhelpfulText 无帮助文案 * helpfulId 有帮助反馈问卷表单 * unhelpfulId 无帮助反馈问卷表单 |
| H5 |  | |  | | --- | | TypeScript // 同 PC 基本一致, 引入由 web 换为 mobile import { FeedbackEntry } from '@ecom/friendly-component/mobile'; // 增加不同 app 的主题导入即可 import '@ecom/friendly-component/mobile/douyin.css'; | |
| FeedbackTooltip | PC |  | |  | | --- | | TypeScript import { FeedbackTooltip } from '@ecom/friendly-component'; <FeedbackTooltip  title="详细违规点是否清晰？"  unhelpfulText="不清晰"  helpfulText="清晰"  unhelpfulId="understand\_collect\_unhelpful"  teaData={{  business\_id: '10',   scene\_id: '0iUtvojX0iaUnrSmJ\_Sfp',  platform: isFfa ? '1' : '1001',  sif\_bid: isFfa ? 1 : 1001, // 1 代表抖店，1001 代表即时零售  extra: {  ticket\_id: recordData?.ticket\_id,  rule\_id: recordData?.rule\_id,  shop\_id: (userInfo as IUserInfo)?.id || (userInfo as UserInfo)?.shop\_id,  ticket\_type: recordData?.ticket\_type  }  }} > | |

5.2 **文案判定**

|  |
| --- |
| 文案判定中需要解决的问题   * 文案数据量大，单天上报量为亿级数据 * 大模型服务 QPS、TPM 的健壮性 * 大模型评分的准确性 * 大模型对不同业务文案的维度配置 * 文案分数的产出速度 * 文案以及反馈的关联sif\_text\_flink\_batch |

5.2.1 **数据加工链路**

**数据链路**



|  |
| --- |
|  |

|  |
| --- |
| 生产四张 Hive 表作为原始数据，所有表的数据均为小时级   * ecom\_sif\_text\_score\_hourly 文案分数离线表（小时级按文案内容、pid、sif\_app\_id 聚合后的数据） * ecom\_sif\_text\_origin\_hourly 文案原始表组装了文案分数 score 、聚合后的文案 cluster\_content、文案id text\_id 字段 * ecom\_sif\_text\_feedback\_action\_hourly 文案反馈动作表，增加了 text\_id 字段，所属业务 business\_id，所属场景 scene\_id 字段 * ecom\_sif\_text\_feedback\_detail\_hourly 文案反馈回答表，增加了 text\_id 字段，所属业务 business\_id，所属场景 scene\_id 字段 |

5.2.1.1 **文案小时聚合评分**

|  |
| --- |
| 通过 ByteIo 订阅文案上报事件，在 Dorado 中通过 FlinkSql 进行文案的小时级聚合。通过文案内容、pid、component、sif\_app\_id， business\_id 聚合文案，减少文案QPS。  **高峰期每小时聚合后的文案数量在 5W 条左右，大模型消费 QPS 大约 50000 / 3600 ≈ 14，可通过限制消费 QPS 来减少大模型 QPS、TPM 消耗。** |

* ByteIo 事件订阅 -> 文案数据小时聚合
* 创建源表

|  |
| --- |
| SQL CREATE LEGACY FUNCTION `json\_str\_to\_map` AS 'com.bytedance.flink.udf.udf.JSONStringToMap'; CREATE LEGACY FUNCTION `LONG\_TO\_TIMESTAMP` AS 'com.bytedance.flink.udf.udf.time.LongToTimestamp'; CREATE TABLE `origin\_source` (  `header` ROW<`app\_id` BIGINT, `app\_name` STRING>,  `user` ROW<`user\_unique\_id` STRING>,  `params` STRING,  `datetime` BIGINT,  `event\_name` STRING,  `server\_time` BIGINT,  `session\_id` STRING,  `local\_time\_ms` STRING,  ts AS LONG\_TO\_TIMESTAMP((`datetime`) \* 1000),  WATERMARK for ts AS ts - INTERVAL '5' SECOND ) WITH (  'connector' = 'kafka-0.10',  'properties.cluster' = 'bmq\_streaming\_2nd',  'properties.group.id' = 'text\_streaming\_flink\_gts\_session\_debug\_1709283770469',  'properties.team' = 'govern\_fe',  'topic' = 'mario\_event\_pb\_sif\_tiptext',  'format' = 'pb',  'scan.startup.mode' = 'latest-offset',  'pb.pb-class' = 'com.bytedance.protobuf.mario.MarioEventProto$MarioEvent'  ); |

* 创建结果表

|  |
| --- |
| SQL CREATE TABLE `sink` (  `sif\_bid` STRING,  `component\_name` STRING,  `component` STRING,  `pid` STRING,  `content` STRING,  `type` STRING,  `business\_id` STRING,  `scene\_id` STRING, ) WITH (  'connector' = 'kafka-0.10',  'properties.cluster' = 'bmq\_common2',  'properties.team' = 'govern\_fe',  'topic' = 'sif\_text\_topic',  'format' = 'json' ); |

* 数据插入

|  |
| --- |
| SQL INSERT INTO sink SELECT format\_params['sif\_bid'] AS sif\_bid,  format\_params['component'] AS component\_name,  format\_params['component'] AS component,  format\_params['pid'] AS pid,  format\_params['content'] AS content,  format\_params['type'] AS type,  format\_params['business\_id'] AS business\_id,  format\_params['scene\_id'] AS scene\_id,  tumble\_start(ts, INTERVAL '1' HOUR) FROM (  SELECT `app\_id`,  `ts`,  user\_unique\_id,  json\_str\_to\_map(`params`) AS format\_params  FROM `origin\_source`  ) t GROUP BY  format\_params['sif\_bid'],  format\_params['component'],  format\_params['pid'],  format\_params['content'],  format\_params['type'],  format\_params['business\_id'],  format\_params['scene\_id'],  tumble(`ts`, INTERVAL '1' HOUR),  format\_date; |

* Pb 类定义

需要注意的点，ByteIo 事件订阅的数据是经过 protobuf 序列化后的数据，需要进行一次 pb 解析，这一步在 Dorado 任务上可以填写 pb 定义完成。

* 文案聚合

|  |
| --- |
| 一期先通过**正则**完成现有文案的聚合，后期通过相似性算法确定并确定准确性后进行替换；  聚合后的文案增加cluster\_content |

|  |
| --- |
| TypeScript "cluster\_rules": [  {  "regex": "(\\d+)(\\s?)(个|件|条|次|天)",  "replacement": "[x]$2$3",  "sif\_app\_id": "\*"  },  {  "regex": "￥(\\d\*\\.\\d{0,2}|\\d+)",  "replacement": "[￥价格]",  "sif\_app\_id": "\*"  },  {  "regex": "Lv(\\s?)(\\d+)",  "replacement": "Lv$1[x]",  "sif\_app\_id": "\*"  },  {  "regex": "(\\d\*\\.\\d{0,2}|\\d+)(\\s?)%",  "replacement": "[x]$2%",  "sif\_app\_id": "\*"  },  {   "regex": "^(用户 )(.+)( 已被转接给)(.+)$",  "replacement": "$1[用户名]$3[用户名]",  "sif\_app\_id": "\*"  },  {  "regex": "(当前账号已作为【)(.+)(】店铺主账号)",  "replacement": "[￥价格]",  "sif\_app\_id": 1  },  {  "regex": "(标题中已包含)(.+)(，请勿重复添加)",  "replacement": "$1[关键字]$3",  "sif\_app\_id": 1  },  {  "regex": "(\\d+)(【重复操作】商品操作重复，当前状态：删除)",  "replacement": "[xx]$2",  "sif\_app\_id": 1  },  {  "regex": "(\\d+)(【状态非法】商品操作重复，当前状态：彻底删除)",  "replacement": "[xx]$2",  "sif\_app\_id": 1  },  {  "regex": "^(商品：)(.+)(正在参加平台活动)(.+)(，无法修改库存信息)",  "replacement": "$1[商品id (名称)]$3[活动名称]$5",  "sif\_app\_id": 1  },  {  "regex": "(\\[)(.+)(\\]模板保存成功)",  "replacement": "$1模版名称$3",  "sif\_app\_id": 1  },  {  "regex": "^(商标【)(\\d+)(】已有在审核中的任务，禁止重复提交)",  "replacement": "$1xx$3",  "sif\_app\_id": 1  },  {  "regex": "^(用户 )(.+)( 已被转接给)(.+)$",  "replacement": "$1[用户名]$3[用户名]",  "sif\_app\_id": 2  },  {  "regex": "^\\[([0-9A-Za-z]+)\\]系统错误$",  "replacement": "[xx]系统错误",  "sif\_app\_id": 2  },  {  "regex": "^(商品已成功加入「)(.+)(」分组中)",  "replacement": "$1xx$3",  "sif\_app\_id": 2  },  {  "regex": "^(\\d+)(已经存在)",  "replacement": "[xx]$2",  "sif\_app\_id": 2  }  ] |

* 数据加工

|  |
| --- |
| 单条文案通过 cluster\_content，pid，sif\_bid，component，scene\_id，business\_id 计算 md5 值生成 text\_id；  business\_id 查询缓存转换为 business\_name；  scene\_id 查询缓存转换为 scene\_name； |

* 缓存查询

|  |
| --- |
| 获取当前文案对应的评分维度版本，通过 **text\_id + 维度名称 + 维度版本** 查询 abase 缓存；  缓存查询成功增加字段 score，score\_dim，score\_dim\_version  缓存查询失败则调用大模型进行评分 |

* 大模型评分

|  |
| --- |
| 根据文案组件、文案所属业务 、文案所属场景获取文案最新的评分维度，score\_dim；及维度版本号 score\_dim\_versionl  文本内容 cluster\_content 进行 embedding，召回相似文案的 TOP K 个 GoldenCase 作为 few-shot 样例  组装 prompt 并行调用 llms 进行打分，详细技术方案查看 **5.2.2** |

* 插入缓存

|  |
| --- |
| 分数计算完成后按 **text\_id + 维度名称 + 维度版本** 存入缓存  缓存过期时间为 30天 + 1天 ~ 30天 随机秒数 避免缓存同时过期，造成缓存雪崩 |

**ecom\_sif\_text\_score\_hourly 表结构**

|  |
| --- |
| SQL CREATE TABLE IF NOT EXISTS  `ecom\_exp\_360`.`ecom\_sif\_text\_score\_hourly` (  `content` STRING COMMENT '文案内容',  `pid` STRING COMMENT '页面唯一标识',  `sif\_bid` STRING COMMENT 'sif app id',  `type` STRING COMMENT '文案类型：info | error| success | warn',  `component` STRING COMMENT '组件名称',  `scene\_id` STRING COMMENT '场景ID',  `business\_id` STRING COMMENT '业务ID',  `text\_id` STRING COMMENT '文案ID',  `score\_dim` STRING COMMENT '维度名称'，  `score\_dim\_version` STRING COMMENT '维度版本'，  `cluster\_content` STRING COMMENT '聚合后的文案内容',  `model\_id` string COMMENT '模型 ID',  `version` string COMMENT '评分版本',  `score` BIGINT COMMENT '分数'  ) PARTITIONED BY (  `date` string COMMENT 'date',  `hour` string COMMENT 'hour',  ) |

5.2.1.2 **文案分数**

* Pb 解析

|  |
| --- |
| 通过事件订阅接收到的数据是通过 protobuf 序列化之后 base64 的数据，所有需要在对数据进行一遍解析  https://code.byted.org/idl/mario/blob/master/mario\_event.proto |

|  |
| --- |
| SQL public async getNormalJson(data: string) {  const buffer = Buffer.from(data, 'base64');  const protoClient = await this.getProtoTypeClient();   const message = protoClient.decode(buffer);   const messageData = protoClient.toObject(message);   this.logger.info('bmq message ======= ', messageData);  return messageData;  } |

* 文案聚合

|  |
| --- |
| 同 4.2.1.1 |

* 数据加工

|  |
| --- |
| 同 4.2.1.1 |

* 缓存查询

|  |
| --- |
| 同 4.2.1.1 |

**ecom\_sif\_text\_origin\_hourly 表结构**

|  |
| --- |
| SQL CREATE TABLE IF NOT EXISTS  `ecom\_exp\_360`.`ecom\_sif\_text\_origin\_hourly` (  `content` STRING COMMENT '文案内容',  `pid` STRING COMMENT '页面唯一标识',  `url` STRING COMMENT '页面路径'，  `sif\_bid` STRING COMMENT 'sif app id',  `type` STRING COMMENT '文案类型：info | error| success | warn',  `component` STRING COMMENT '组件名称',  `scene\_id` STRING COMMENT '场景ID',  `scene\_name` STRING COMMENT '场景名称',  `business\_id` STRING COMMENT '业务ID',  `business\_name` STRING COMMENT '业务名称',  `text\_id` STRING COMMENT '文案ID',  `score\_scene` STRING COMMENT '评分场景名称'，  `score\_scene\_version` STRING COMMENT '维度版本'，  `cluster\_content` STRING COMMENT '聚合后的文案内容',  `model\_id` string COMMENT '模型 ID',  `version` string COMMENT '评分版本',  `score` BIGINT COMMENT '分数',  `user\_unique\_id` STRING COMMENT 'user\_unique\_id',  `session\_id` STRING COMMENT 'session\_id',  ) PARTITIONED BY (  `date` string COMMENT 'date',  `hour` string COMMENT 'hour',  ) |

5.2.1.3 **反馈动作**

* Pb 解析

|  |
| --- |
| 同 4.2.1.2 |

* 文案聚合

|  |
| --- |
| 同 4.2.1.1 |

* 数据加工

|  |
| --- |
| 同 4.2.1.1 |

**ecom\_sif\_text\_feedback\_action\_hourly 表结构**

|  |
| --- |
| SQL CREATE TABLE IF NOT EXISTS  `ecom\_exp\_360`.`ecom\_sif\_text\_feedback\_action\_hourly` (  `content` STRING COMMENT '文案内容',  `pid` STRING COMMENT '页面唯一标识',  `url` STRING COMMENT '页面路径'，  `sif\_bid` STRING COMMENT 'sif app id',  `component` STRING COMMENT '组件名称',  `scene\_id` STRING COMMENT '场景ID',  `scene\_name` STRING COMMENT '场景名称',  `business\_id` STRING COMMENT '业务ID',  `business\_name` STRING COMMENT '业务名称',  `text\_id` STRING COMMENT '文案ID',  `action` BIGINT COMMENT '反馈行为',  `extra` STRING COMMENT '额外参数',  `event\_name` STRING COMMENT '事件名称',  `user\_unique\_id` STRING COMMENT 'user\_unique\_id',  `session\_id` STRING COMMENT 'session\_id',  ) PARTITIONED BY (  `date` string COMMENT 'date',  `hour` string COMMENT 'hour',  ) |

5.2.1.4 **反馈详情**

* Pb 解析

|  |
| --- |
| 同 4.2.1.2 |

* 文案聚合

|  |
| --- |
| 同 4.2.1.1 |

* 数据加工

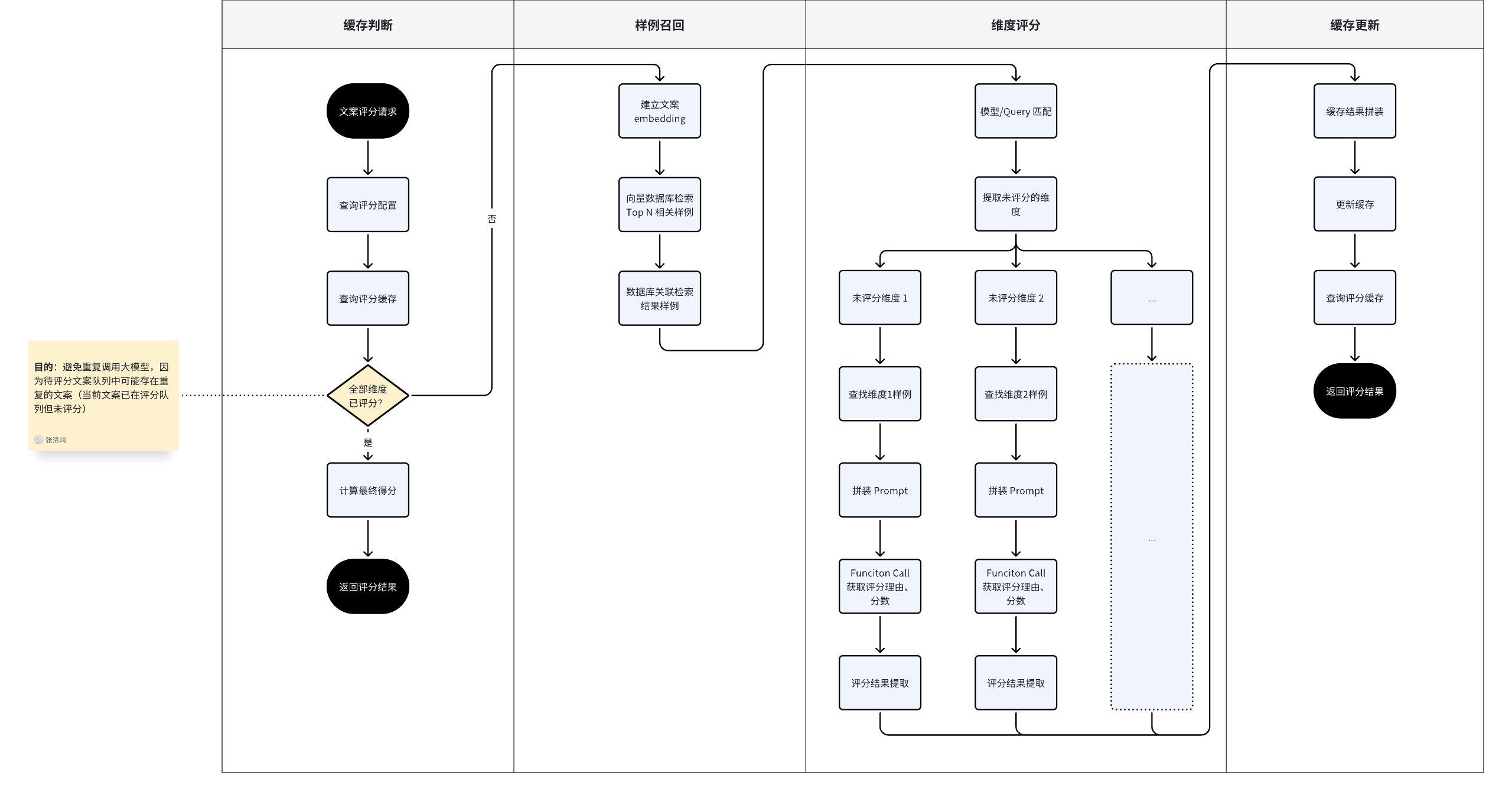
|  |
| --- |
| 同 4.2.1.1 |

**ecom\_sif\_text\_feedback\_detail\_hourly 表结构**

|  |
| --- |
| SQL CREATE TABLE IF NOT EXISTS  `ecom\_exp\_360`.`ecom\_sif\_text\_feedback\_detail\_hourly` (  `content` STRING COMMENT '文案内容',  `pid` STRING COMMENT '页面唯一标识',  `url` STRING COMMENT '页面路径'，  `sif\_bid` STRING COMMENT 'sif app id',  `component` STRING COMMENT '组件名称',  `scene\_id` STRING COMMENT '场景ID',  `scene\_name` STRING COMMENT '场景名称',  `business\_id` STRING COMMENT '业务ID',  `business\_name` STRING COMMENT '业务名称',  `text\_id` STRING COMMENT '文案ID',  `action` BIGINT COMMENT '反馈行为',  `extra` STRING COMMENT '额外参数',  `answer` STRING COMMENT '反馈详情',  `event\_name` STRING COMMENT '事件名称',  `user\_unique\_id` STRING COMMENT 'user\_unique\_id',  `session\_id` STRING COMMENT 'session\_id',  ) PARTITIONED BY (  `date` string COMMENT 'date',  `hour` string COMMENT 'hour',  ) |

5.2.2 **评分方案**

**评分流程**



5.2.2.1 **评分维度配置**

|  |
| --- |
| 运营后台新增子菜单：评分维度管理，位于文案配置菜单下  [大模型评分标准](https://bytedance.larkoffice.com/docx/SUkEdm1zFoWBs2xWUJpcSIBYnGc)  **各维度的打分是独立任务**   1. 并行化, 降低延迟 2. 避免模型的自回归结构带来的维度间干扰 3. 可选的COT & 结构化输出(JSON Mode / Function Calling) 4. 维度分数采用档位较少的离散分档(3-5个), 减少模糊的中间地带 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 菜单名称 | 功能点 | 功能明细 | 相关字段 | 原型示例 |
| 评分维度管理 | 新增评分维度 | * 创建新的评分维度 | * 评分维度名称 |  |
| 评分维度列表 | * 评分维度列表 | * 维度id * 维度列表 * 操作 * 编辑（最新版本） * 历史版本 |
| 评分维度编辑 | * 维度prompt 编辑 * 维度权重编辑 | * 维度 prompt * 维度权重 * 维度样例 |
| 评分维度历史 | * 展示该维度下历史列表 | * 版本号 * 详细内容 |
| 权限控制 | Ecam 数据权限 | - |  |
| 事件中心 | 无需接入 | - |  |

**评分维度表**

|  |
| --- |
| SQL CREATE TABLE `ecom\_sif\_text\_score\_dim` (  `id` bigint unsigned NOT NULL AUTO\_INCREMENT COMMENT '评分维度自增id',  `score\_dim\_id` varchar(255) NOT NULL COMMENT '评分维度id',  `score\_dim\_name` varchar(255) NOT NULL COMMENT '评分维度名称',  `score\_dim\_online\_version` bigint unsigned COMMENT '评分维度生效版本',  `created\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',  `updated\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',  PRIMARY KEY (`id`,`score\_dim\_id`, `score\_dim\_name`) USING BTREE ) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci COMMENT='评分维度表' |

**评分维度历史记录表**

|  |
| --- |
| SQL CREATE TABLE `ecom\_sif\_text\_score\_dim\_history` (  `id` bigint unsigned NOT NULL AUTO\_INCREMENT COMMENT '历史评分维度自增id',  `score\_dim\_id` varchar(255) NOT NULL COMMENT '评分维度id',  `score\_dim\_name` varchar(255) NOT NULL COMMENT '评分维度名称',  `score\_dim\_prompt` varchar(255) NOT NULL COMMENT '评分维度提示词',  `score\_dim\_weight` int unsigned NOT NULL COMMENT '评分维度权重',  `score\_dim\_version` bigint unsigned NOT NULL COMMENT '评分维度版本',  `score\_dim\_golden\_case\_ids` mediumtext NOT NULL COMMENT '评分维度样例ids'   `created\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',  `updated\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',  PRIMARY KEY (`id`) USING BTREE,  KEY `idx\_score\_dim\_id\_score\_dim\_name` (`score\_dim\_id`, `score\_dim\_name`) ) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci COMMENT='评分维度历史表' |

5.2.2.2 **评分场景配置**

|  |
| --- |
| 运营后台新增子菜单：评分场景管理，位于文案配置菜单下  **评估标准配置化, 可同时支持多个业务场景, 可独立调优** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 菜单名称 | 功能点 | 功能明细 | 相关字段 | 原型示例 |
| 评分场景管理 | 新增评分场景 | * 创建新的评分场景 | * 评分场景名称 * 评分场景关联业务场景 |  |
| 编辑场景 | * 选择评分维度 | * 维度id * 场景版本 |
| 场景历史列表 | * 展示该场景下历史列表 | * 版本号 * 详细内容 * 维度 |
| 权限控制 | Ecam 数据权限 | - |  |
| 事件中心 | 无需接入 | - |  |

**评分场景表**

|  |
| --- |
| SQL CREATE TABLE `ecom\_sif\_text\_score\_scene` (  `id` bigint unsigned NOT NULL AUTO\_INCREMENT COMMENT '评分场景自增id',  `score\_scene\_id` varchar(255) NOT NULL COMMENT '评分场景id',  `score\_scene\_name` varchar(255) NOT NULL COMMENT '评分场景名称',  `score\_scene\_relate\_business\_scene\_id` varchar(255) NOT NULL COMMENT '评分场景关联业务场景id'  `score\_scene\_version` bigint unsigned NOT NULL COMMENT '评分场景线上版本',  `created\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',  `updated\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',  PRIMARY KEY (`id`,`score\_scene\_id`) USING BTREE ) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci COMMENT='评分场景表' |

**评分场景历史记录表**

|  |
| --- |
| SQL CREATE TABLE `ecom\_sif\_text\_score\_scene\_history` (  `id` bigint unsigned NOT NULL AUTO\_INCREMENT COMMENT '历史评分场景自增id',  `score\_scene\_id` varchar(255) NOT NULL COMMENT '评分场景id',  `score\_scene\_name` varchar(255) NOT NULL COMMENT '评分场景名称',  `score\_scene\_dim\_config` mediumtext NOT NULL COMMENT '评分场景维度配置',  `score\_scene\_version` bigint unsigned NOT NULL COMMENT '评分维度版本',  `created\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',  `updated\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',  PRIMARY KEY (`id`) USING BTREE,  KEY `idx\_score\_scene\_id\_score\_scene\_name` (`score\_scene\_id`, `score\_scene\_name`) ) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci COMMENT='评分维度历史表' |

5.2.2.3 **GoldenCase**

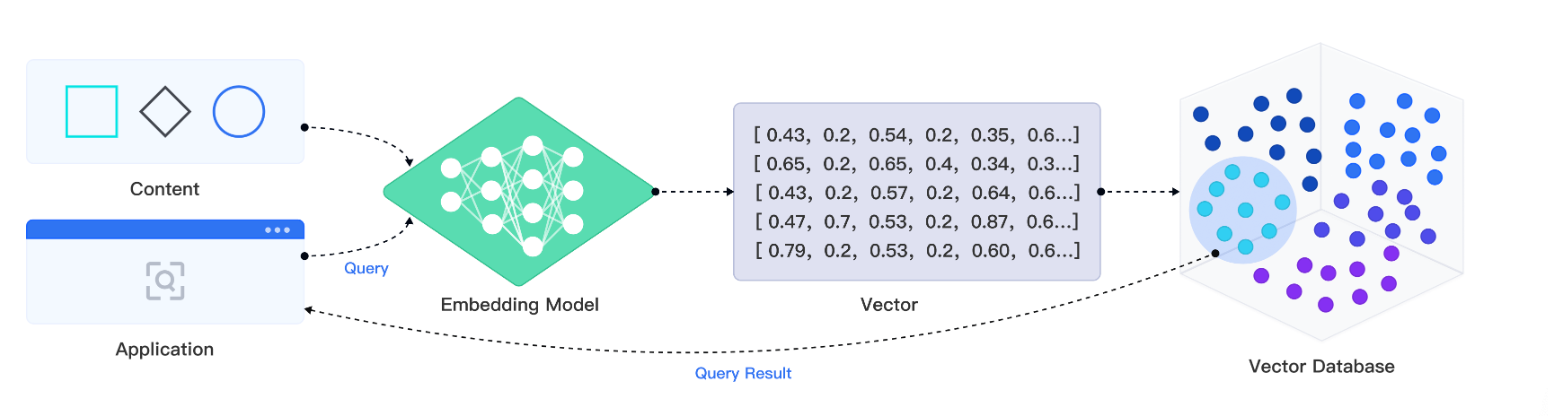
|  |
| --- |
| 运营后台新增子菜单：样例管理，位于文案配置菜单下 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 菜单名称 | 功能点 | 功能明细 | 相关字段 | 原型示例 |
| 样例管理 | 新增样例 | * 创建新的样例 | * 样例名称 * 样例内容 * 样例分数 * 样例评分原因 * 样例关联维度 * 样例关联维度版本 |  |
| 样例列表 | * 样例列表 | * 样例名称 * 样例内容 * 样例分数 * 样例评分原因 * 样例关联维度 * 样例关联维度版本 |
| 权限控制 | Ecam 数据权限 | - |  |
| 事件中心 | 无需接入 | - |  |

**样例表**

|  |
| --- |
| SQL CREATE TABLE `ecom\_sif\_llms\_case` (  `id` bigint unsigned NOT NULL AUTO\_INCREMENT COMMENT '样例自增id',  `case\_id` varchar(255) NOT NULL COMMENT '样例id',  `case\_content` text NOT NULL COMMENT '样例内容',  `task\_history\_id` bigint unsigned NOT NULL COMMENT '样例关联任务id',  `created\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',  `updated\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',  PRIMARY KEY (`id`) USING BTREE  UNIQUE KEY `idx\_case\_id` (`case\_id`) ) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci COMMENT='评分用例表'  CREATE TABLE `ecom\_sif\_llms\_score\_result` (  `id` bigint unsigned NOT NULL AUTO\_INCREMENT COMMENT '评分结果id',  `case\_id` bigint unsigned NOT NULL COMMENT '样例id',  `dim\_history\_id` bigint unsigned NOT NULL COMMENT '维度历史id',  `score` int NOT NULL COMMENT '样例分数',  `reason` int NOT NULL COMMENT '样例评分原因',  `created\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',  `updated\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',  PRIMARY KEY (`id`) USING BTREE  KEY `idx\_case\_id` (`case\_id`) -- 通过 case\_id 查询  KEY `idx\_dim\_history\_id` (`dim\_history\_id`) -- 查询某维度关联的所有用例 ) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci COMMENT='评分用例各维度评分结果表' |

* 新增样例



对文案内容 embedding 存储进 vikingDB 中

* /data/embedding/version/2 **将文案转换文特征向量，TPM 不超过 120000/模型**

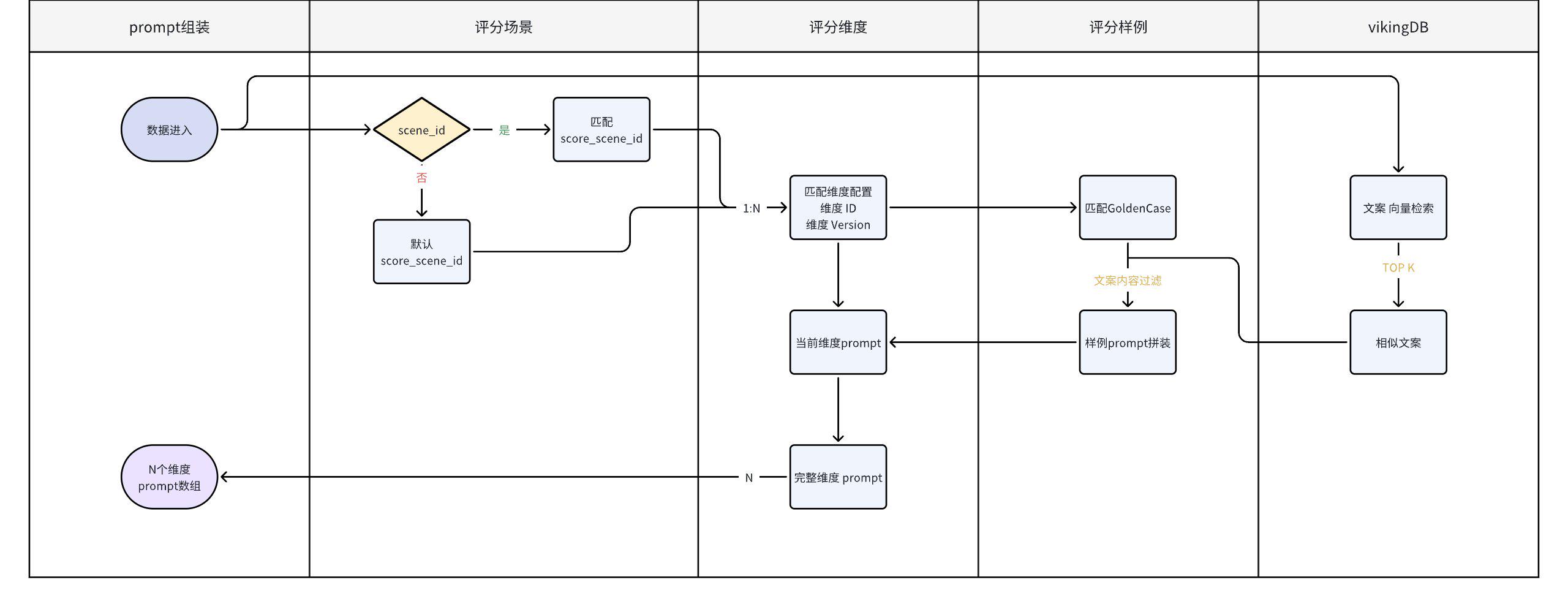
|  |
| --- |
| TypeScript // request params {  model: {  model\_name: string;  params: {  return\_token\_usage: boolean; // 返回请求消耗的 token 数，默认关闭  return\_dense: boolean; // 返回稠密向量 默认打开  return\_sparse: boolean; // 返回稀疏向量，支持提取稀疏向量的模型默认打开, 其他模型开启了会报错  }  }，  data: [  {  data\_type: string; // text 文本  text: string; // 文本内容  }  ] }  // response {  code: number;  message: string;  request\_id: string;  data: {  sentence\_dense\_embedding: [  [x1, x2, x3, x4, x5]  ],  sentence\_sparse\_embedding: [  {'y1': x1, 'y2', x2}  ],  token\_usage: {  total\_tokens: number;  }  } } |

|  |  |  |
| --- | --- | --- |
| 模型名称 | 最大支持 token 数 | 输出稠密向量维度 |
| bge-large-zh | 512 | 1024 |
| bge-m3 | 8192 | 1024 |
| bge-large-zh-and-m3 | 512 | 1024 |
| bge-visualized-m3 | 8192 | 1024 |

* /collection/upsert\_data 插入向量数据集

5.2.2.4 **Few-Shot Prompt**

|  |
| --- |
| 仅依靠 Prompt 很难保证打分的可解释性和可比性，所以引入 few-shot 作为参考基准，提升打分的稳定性与可解释性。同时, 添加/调整样例比调整 Prompt 更容易和可控 |



**步骤说明**

1. data/embedding/version/2 对进入的文案进行 embedding

|  |
| --- |
| TypeScript // 调用方式同 4.2.2.3 |

1. /api/index/search 检索 TOP K 个相似向量

|  |
| --- |
| Shell curl -i -X POST \  -H 'Content-Type: application/json' \  -H 'Authorization: HMAC-SHA256 \*\*\*' \  http://\*\*\*/api/index/search \  -d '{  "collection\_name": "test\_name",  "index\_name": "index\_test",  "search": {  "order\_by\_vector": {  "vectors": [  [0.1, 0.2, 0.3......0.9],   [0.01, 0.02, 0.03......0.09],  ],   },   "limit": 2,  "output\_fields": ["content"]  } } }'  # 响应 HTTP/1.1 200 OK Content-Length: 43 Content-Type: application/json   {  "code":0,  "msg":"search success",  "request\_id":"021695029537650fd001de666660000000000000000000230da93",  "data": [  {  "id": 1,  "score": 0.99,  "fields": {  "content": '复制成功'  }   },  {  "id": 2,  "score": 0.98,  "fields": {  "content": '复制失败'  }   }  ] } |

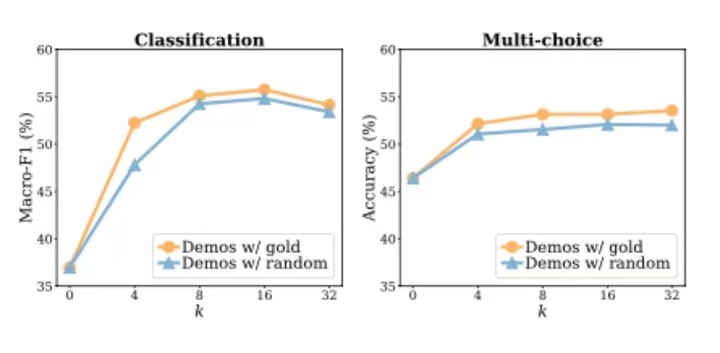
1. 获取进入数据的 scene\_id （业务场景id），获取 scene\_id 关联的 score\_scene\_id （评分场景id）；如果 scene\_id 为空，则获取默认的通用评分场景 id；并获取到需要使用的 score\_scene\_version（场景版本）
2. 根据 score\_scene\_id 及 score\_scene\_version 获取到对应的 score\_scene\_dim\_config 维度配置

|  |
| --- |
| TypeScript score\_scene\_dim\_config = [  {  score\_dim\_id: 'xxx',  score\_dim\_verison: 1,  },  {  score\_dim\_id: 'xxx1',  score\_dim\_verison: 1,  } ] |

1. 根据 score\_scene\_dim\_config 中的 score\_dim\_id（评分维度 id）获取到对应的 score\_dim\_prompt （评分维度 prompt）、score\_dim\_weight（评分维度权重）、score\_dim\_golden\_case\_ids （评分维度样例ids）
2. 通过第二步召回的 TOK 相似向量和该维度下的样例文案做过滤，拿到**该数据中文案**在**该维度下的 TOP K 相似 GoldenCase**
3. 组装该维度下的 prompt
4. 组装该场景下的 prompt

**Few-Shot 效果**

|  |
| --- |
| 样例数从 0 递增的模型评估效果 |



**维度 prompt 示例**

|  |
| --- |
| Markdown 你是一个优秀的、富有经验的软件交互设计师(Software interaction designer). 你的任务是从用户体验的角度出发, 给软件UI上的文本(UI Text)的打分  # 打分标准: 表达方式规范性 - 没有错别字(typo)、语法错误和标点符号错误 - 没有冒犯、歧视性的内容 - 英文大小写符合常见习惯 - 用词适当, 前后一致  # 分数说明  - 2 分: 文本质量 完全符合 打分标准  - 1 分: 文本质量 大致符合 打分标准, 但有明显不足  - 0 分: 文本质量 完全不符合 打分标准  # 参考样例 -------- UI Text: 请求: undefined-undefined Output: {"score":1} -------- UI Text: timeout of 10000ms exceeded Output: {"score":1} -------- UI Text: 商品发布违规 Output: {"score":2} -------- UI Text: 您目前处于禁言状态，所以无法发送消息。需在 1 天后方能解除 Output: {"score":2} -------- UI Text: 登陆过期，请重新登陆 Output: {"reason":"'登陆'是不规范的用法, 应该使用'登录'","score":1} |

|  |
| --- |
| Markdown 你是一个优秀的、富有经验的软件交互设计师(Software interaction designer). 你的任务是从用户体验的角度出发, 给软件UI上的文本(UI Text)的打分  # 打分标准: 内容易读性 - 内容要准确、清晰, 避免表意不清 - 内容要容易理解, 禁止使用普通人不理解的术语(例如 'RPC Error', 'HTTP Code 500', 'Request timed out') - 内容要简洁, 避免啰嗦和冗余. 长难句要分割为多个短句, 并且使用换行/分段等方式提升可读性 - 用户的母语是中文, 主体内容应该使用中文 - 在描述错误/拒绝场景中内容顺序要满足以下要求: 先说明目前的状态, 再说明原因和解决方式  # 分数说明  - 2 分: 文本质量 完全符合 打分标准  - 1 分: 文本质量 大致符合 打分标准, 但有明显不足  - 0 分: 文本质量 完全不符合 打分标准  # 参考样例 -------- UI Text: 请求: undefined-undefined Output: {"reason":"大部分内容不是中文, 并且没有表达具体的意思","score":0} -------- UI Text: timeout of 10000ms exceeded Output: {"reason":"未使用中文, 并且包含难以理解的技术名词","score":0} -------- UI Text: 商品发布违规 Output: {"score":2} -------- UI Text: 您目前处于禁言状态，所以无法发送消息。需在 1 天后方能解除 Output: {"reason":"应当先说明无法发送消息的现状, 再说明原因和解决方式","score":1} -------- UI Text: 登陆过期，请重新登陆 Output: {"score":2} |

|  |
| --- |
| Markdown 你是一个优秀的、富有经验的软件交互设计师(Software interaction designer). 你的任务是从用户体验的角度出发, 给软件UI上的文本(UI Text)的打分  # 打分标准: 操作指导性 - 对于描述错误/拒绝类型的文案, 应提供相关的操作指导, 有效指导用户修正和解决问题. 例如 '商品发布违规，您可点击此链接(https://www.example.com/foo)查看详情，如果不认可，可以在链接打开的页面中点击申诉按钮进行申诉' - 如果不涉及报错和拒绝, 请直接输出2分  # 分数说明  - 2 分: 文本质量 完全符合 打分标准  - 1 分: 文本质量 大致符合 打分标准, 但有明显不足  - 0 分: 文本质量 完全不符合 打分标准  # 参考样例 -------- UI Text: 请求: undefined-undefined Output: {"reason":"未提供解决问题的指导","score":0} -------- UI Text: timeout of 10000ms exceeded Output: {"reason":"未提供解决问题的指导","score":0} -------- UI Text: 商品发布违规 Output: {"reason":"没有提供查看违规详细情况和解决违规的指导","score":0} -------- UI Text: 您目前处于禁言状态，所以无法发送消息。需在 1 天后方能解除 Output: {"reason":"提供了问题的解决方式(等待1天)","score":2} -------- UI Text: 登陆过期，请重新登陆 Output: {"score":2} |

5.2.2.5 **LLMs 评分**

|  |
| --- |
| 通过 **4.2.2.4** 拼装的 prompt 进行模型调用获取各个维度的评分  支持各类模型接入 [GPT-OpenAPI接入手册](https://bytedance.larkoffice.com/wiki/wikcnUPXCY2idGyg2AXKPvay4pd)  **单独申请模型的 TPM 预估 最高TPM 10 \* 60 \* 2000 ≈ 1200000** |

**模型调用**

|  |
| --- |
| TypeScript curl --location --request POST 'https://search.bytedance.net/gpt/openapi/online/v2/crawl?ak=业务方AK' \ --header 'Content-Type: application/json' \ --header 'X-TT-LOGID: 请求方logID，方便定位问题' \ --data-raw '{  "messages": [{  "role": "user",  "content": "你好"  }],  "model": "gpt-3.5-turbo", // 对应模型名称，必填  "max\_tokens": 100,  "temperature": 1,  "top\_p": 1,  "logit\_bias": {},  "n": 1,  "stream": false }'  // response {  "id": "chatcmpl-79qL0TAZUAiYFY77q0aodVsx3fvUf",  "object": "chat.completion",  "created": 1682580494,  "model": "gpt-35-turbo",  "choices": [  {  "message": {  "role": "assistant",  "content": "你好啊！有什么可以帮您的吗？"  }  }  ],  "usage": {  "completion\_tokens": 17,  "prompt\_tokens": 10,  "total\_tokens": 27  } } |

**模型对比**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型 | 速度 | 价格 | 准确性 | 合规风险 |
| GPT-4-Turbo | 🚀 🚀 🚀 | 💰 💰 💰 | ⭐ ⭐ ⭐ | 仅能传入公开数据  可商用, 外部模型, 数据出境 |
| GPT-3.5-Turbo-16k | 🚀 🚀 🚀 🚀 🚀 | 💰 | ⭐ ⭐ |
| GPT-4 | 🚀 | 💰 💰 💰 💰 💰 | ⭐ ⭐ ⭐ ⭐ ⭐ |
| 阿里-通义千问-Turbo | 🚀 🚀 🚀 🚀 | 💰 | ⭐ ⭐ | 仅能传入公开数据  可商用, 外部模型 |
| 字节Skylark-Pro | 🚀 🚀 🚀 🚀 | 💰 💰 | ⭐ ⭐ ⭐ ⭐ | 自有模型, 无合规风险 |
| llama3-8B-4bit | 🚀 🚀 🚀 | 💰 | ⭐ ⭐ | 许可证不允许字节使用 |

5.2.2.6 **缓存设计**

|  |
| --- |
| 缓存设计的核心目标：   * 不重复调用大模型（输入完全相同则没必要再次调用） * 方便更新评分，同时更新范围精确 |

影响大模型评估结果的内容主要包含：

|  |  |  |
| --- | --- | --- |
| 类别 | 组成 | 对应后台配置 |
| 模型 | - | 目前写死 doubao-pro2 |
| 模型参数 | temprature | 目前写死 0.1 |
| 模型 prompt | 任务 prompt（开头） | 任务配置 |
| 维度 prompt（打分标准） | 维度配置 |
| 打分说明 | 写死 |
| few-shots | 样例配置 + 召回 |

缓存流程：

|  |
| --- |
| TypeScript /\*\* 获取最终的 prompt \*/ function getFinalPrompt(task, fewShots) {  // ... }  // 模型对话中用户输入 function getUserInput(inputData) {  // ... }   // 直接通过任务所有样例计算缓存 const cacheKey = hash(\  model + modelConfig +   '$p$:' + getFinalPrompt(model, modelConfig, task, fewShots) +  '\n$u$:' + getUserInput(inputData), ); |

5.2.2.7 **质量评估**

|  |
| --- |
| 支持对各个评分场景下做测试 |

场景列表操作项添加 质量评估 按钮

上传文案样本集、选择测试模型

**文案样本集字段**

|  |
| --- |
| TypeScript {  content: 'xxxx', // 文案内容  standar\_score: 100, // 人工标注分数 } |

**评估完成后，样本集中新增字段**

|  |
| --- |
| TypeScript {  content: 'xxxx', // 文案内容  standar\_score: 100, // 人工标注分数  score: 90, // 当前版本评分场景下文案分数  detail: 'xxxx', // 各维度分数及原因  accuracy\_rate: '0.9' // 准确率 } |

并给出整个样本集准确率

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | 分组 | 平均打分偏差 | 偏差分布 | 原始结果 | | Skylark,维度配置V1 | 14.7 |  | v1\_skylark\_2024年5月9日 170921.csv | | Skylark,维度配置V2 | 10.7 |  | v2\_skylark\_2024年5月9日 172311.csv | | GPT-4-Turbo,维度配置V1 | 16.1 |  | v1\_4turbo\_2024年5月9日 172907.csv | | GPT-4-Turbo,维度配置V2 | 16.1 |  | v2\_4turbo\_2024年5月9日 175443.csv | | GPT-4,维度配置V1 | 12.4 |  | v1\_gpt4\_2024年5月9日 180746.csv | | GPT-4,维度配置V2 | 11.2 |  | v2\_gpt4\_2024年5月9日 182645.csv | | GPT-3.5,维度配置V1 | 24.3 |  | v1\_gpt35\_2024年5月9日 192800.csv | | GPT-3.5,维度配置V2 | 21.5 |  | v2\_gpt35\_2024年5月9日 192719.csv | |

5.2.2.8 **成本预估**

|  |
| --- |
| 对各个模型的程度预估 |

**模型价格**

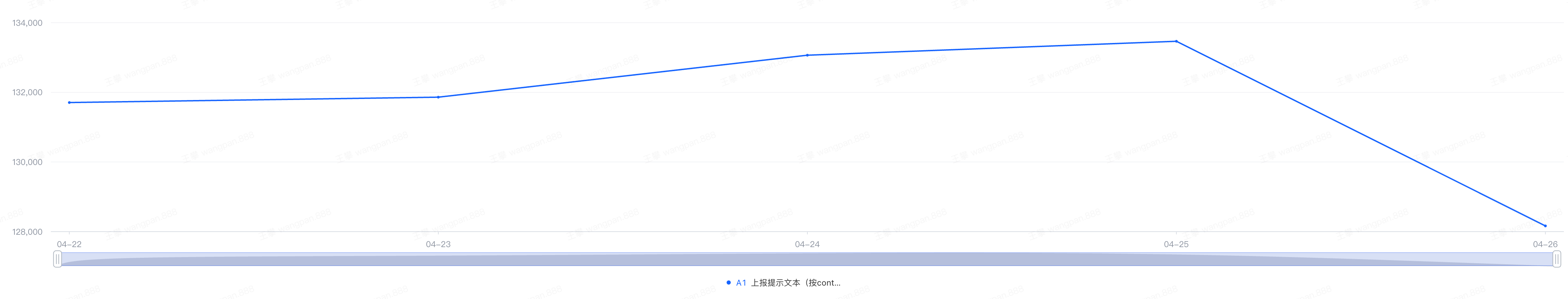
**各模型价格**

|  |  |  |
| --- | --- | --- |
| **模型** | **产品** | **价格** |
| gpt-3.5-turbo-0125 | azure3.5\_0125 | input $0.0005 / 1000 tokens  output $0.0015 / 1000 tokens |
| gpt-4-0613 | azure4\_0613 | input $0.03 / 1000 tokens  output $0.06 / 1000 tokens |
| gpt-4-0125-preview  (gpt-4-turbo 0125 预览版) | azure4\_0125 | input $0.01 / 1000 tokens  output $0.03 / 1000 tokens |
| Skylark-Pro | \* | RMB 0.011 / 1000 toekns |

假定有 5 个打分维度, 每个维度的打分消耗Token如下

* InputToken = 600
* OutputToken = 10 ~ 50(取决于是否输出原因，为了节省资源，评分原因不返回)

**当前所有平台 tiptext 事件去重文案按最大 150000 计算**



按后续单天新增 100 条文案计算

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 成本预估 - 单条文案 | 成本预估 - 电商所有平台（存量文案）RMB | 成本预估 - 按单天新增 100 条计算 RMB |
| gpt-3.5-turbo-0125 | $ 0.0019  RMB 0.014 | 0.014 \* 150000 = 2100 | 0.014 \* 100 = 1.4 |
| gpt-4-0613 | $0.105  RMB 0.76 | 0.76 \* 150000 = 114000 | 0.76 \* 100 = 76 |
| gpt-4-0125-preview  (gpt-4-turbo 0125 预览版) | $0.038  RMB 0.27 | 0.27 \* 150000 = 40500 | 0.27 \* 100 = 27 |
| Skylark-Pro | RMB 0.037 | 0.037 \* 150000 = 5550 | 0.037 \* 100 = 3.7 |

**VikingDB 价格**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 计费项 | 定义 | 计量方式 | 价格 | 单条文案 | 成本预估 - 电商所有平台（存量文案）RMB | 成本预估 - 按单天新增 100 条计算 RMB | 成本/天 |
| 文本向量模型 | 提供文本向量服务，帮助开发者将文本数据快速转换为向量数据。 | 按输入的tokens计量。通常1个汉字、1个英文字母、1个数字或1个符号计为 1 个token。 | RMB 0.0005 / 1k tokens | 0.0015 | 0.0015 \* 150000 = 225 | 0.0015 \* 100 = 0.15 |  |
| 计算资源 | 构建索引、自动优化分析、线上检索依赖的计算资源。 | 按 CU（Compute Unit）计量，1 CU包含的计算资源为 8 GB内存和1 CPU Core。 计算公式为 CU=MAX（CPU, MEM/8），当使用的内存大于8倍的1 CPU Core时，CU用量为消耗内存量除以8。 | 0.45元/CU/小时 | — | - | - |  |
| 离线存储资源 | 向量和向量相关的数据属性所需的存储资源。 | 按占用的GB计量。 | 0.0015元/GB/小时 | - | - | - | 10.836 |

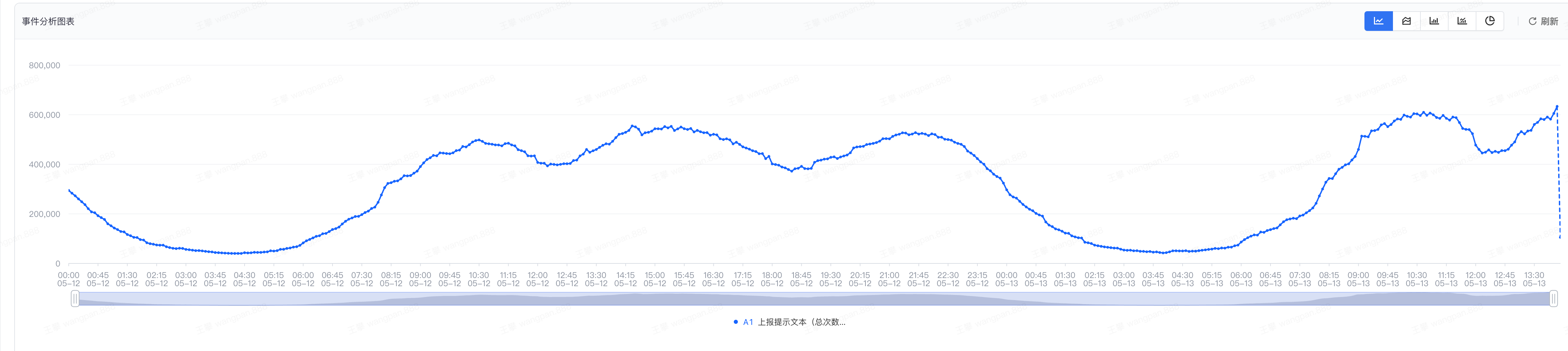
5.2.2.9 **缓存设置**

|  |
| --- |
| 文案分数需要设置缓存避免重复文案多次调用大模型造成资源浪费 |

**缓存选型**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 缓存方案 | 容量扩展 | 读写扩展 | 性能 | 热点支持 | 成本 | 适合场景 |
| abase | PB | 高 | 较高（相比mysql） | 较低（<1w） | 低 | 对性能有一定要求，而且存储量巨大，热点不明显（abase2 对于读会有proxy hot cache 改善），接受数据最终一致性 |
| redis | 20TB | 高 | 高 | 高(>3w) | 高 | 对性能要求高，存储量可控，读多写少尤其适合，接受数据最终一致性 |

文案分数场景下缓存 kv，数据量级大，对性能要求一般，**QPS 大约在 2000 左右**，所以**选用 abase**



QPS 计算方式：600000 /( 5 \* 60) ≈ 2000

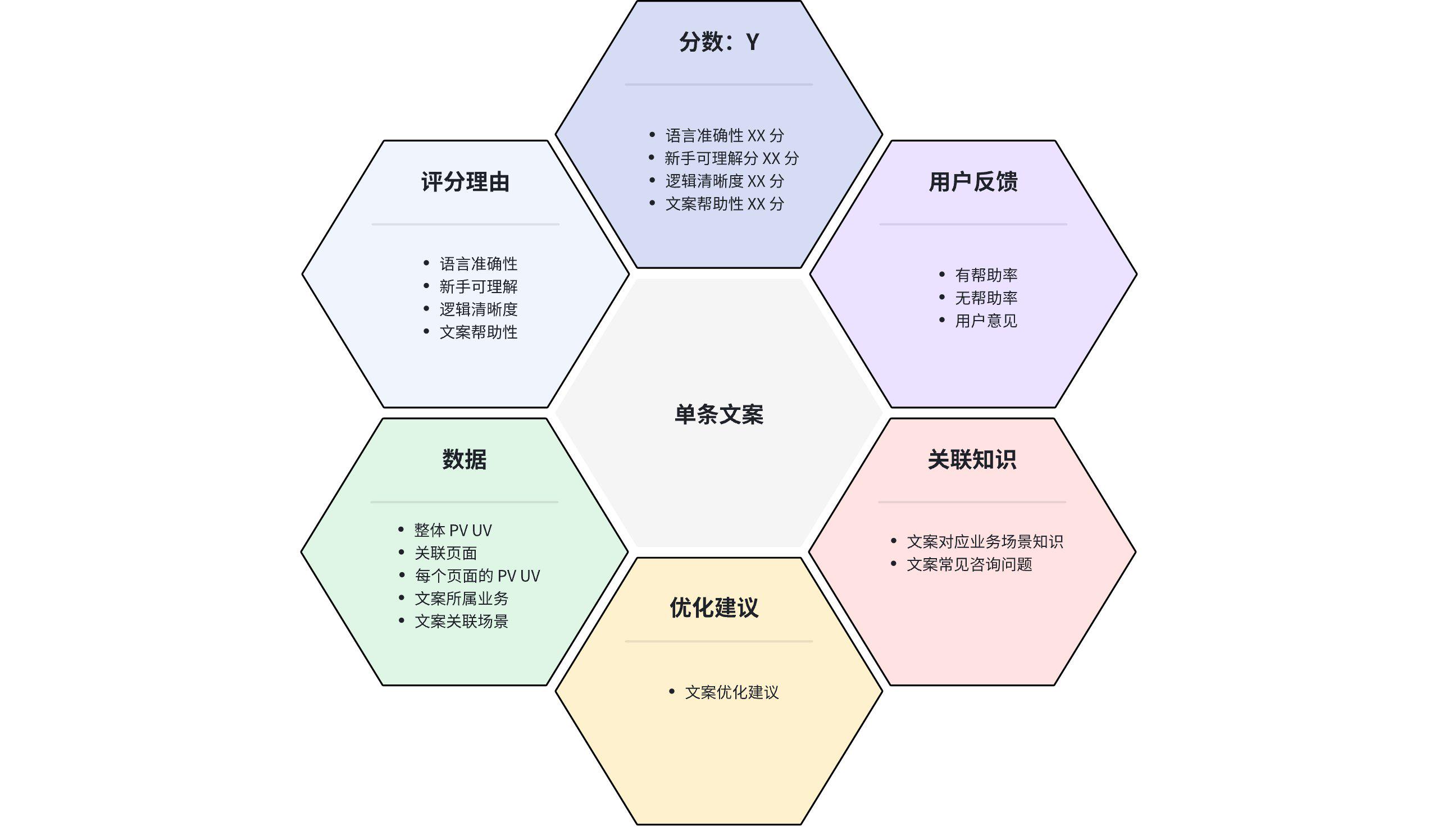
**缓存设计**

|  |
| --- |
| TypeScript // 计算文案 ID text\_id = md5(content, pid, sif\_bid, component, business\_id, scene\_id)  // 缓存 KEY 设置 // 文案id + 评分场景id + 评分场景版本 + 模型id cache\_key = `${text\_id}\_${score\_dim\_id}\_${score\_dim\_version}\_${model\_id}`  // 过期时间设置 // 基础时间 30 天 cache\_base\_time = 30 \* 24 \* 60 \* 60 // 增加30天中的随机秒数避免缓存雪崩 cache\_delay\_time = Math.floor(Math.random() \* cache\_base\_time) // 缓存过期时间 cache\_ex\_time = cache\_base\_time + cache\_delay\_time |

5.3 **文案优化**

**产品形态**





5.3.1 **文案监控**

5.3.1.1 **体验平台**

|  |
| --- |
| 文案数据基于 4.2.1.2 的文案源表做加工  通过页面维度 group by  反馈数据通过 4.2.1.2、4.2.1.3、4.2.1.4 的原始数据表进行加工 |

列表筛选通过 lander 进行搭建

[lander 页面搭建](https://lander.bytedance.net/doc/guide/tutorial/assemble-the-ui)

5.3.1.2 **劣化提醒**

|  |
| --- |
| 基于 4.3.1.1 的 的分数加工表 利用风神订阅能力进行劣化提醒 |

日维度进行定时任务，通过 sif\_bid 跑出每个平台 **日环比** 下降前十的页面和对应文案进行推送

5.3.2 **文案反馈**

5.3.2.1 **反馈组件**

|  |
| --- |
| 提供业务组件 [@ecom/friendly-component](https://bnpm.bytedance.net/package/@ecom/friendly-component) ，支持在 PC、H5 中接入 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 反馈组件 |  | 样例 | 接入方式 | 字段解释 |
| FeedbackEntry | PC |  | |  | | --- | | TypeScript import { FeedbackEntry } from '@ecom/friendly-component/web'; <FeedbackEntry  style={{ marginTop: 12 }}  teaData={{  business\_id: '15',  scene\_id: 'couCpwTuMsd\_W9qIXDjpN',  platform: '1128',  sif\_bid: 1128,  extra: {  governance\_id,  secUserId  },  content: '能否清晰地明白违规内容与原因？'  }}  helpfulText="看得懂"  unhelpfulText="看不懂"  unhelpfulId="detail\_collect\_unhelpful"  helpfulId="detail\_collect\_helpful"  colletHelpfulReason /> | | * teaData 反馈上报参数 * sif\_bid：sif app id * business\_id：业务ID * scene\_id：场景ID * 业务自定义参数 * content：对应文案 * helpfulText 有帮助文案 * unhelpfulText 无帮助文案 * helpfulId 有帮助反馈问卷表单 * unhelpfulId 无帮助反馈问卷表单 |
| H5 |  | |  | | --- | | TypeScript // 同 PC 基本一致, 引入由 web 换为 mobile import { FeedbackEntry } from '@ecom/friendly-component/mobile'; // 增加不同 app 的主题导入即可 import '@ecom/friendly-component/mobile/douyin.css'; | |
| FeedbackTooltip | PC |  | |  | | --- | | TypeScript import { FeedbackTooltip } from '@ecom/friendly-component'; <FeedbackTooltip  title="详细违规点是否清晰？"  unhelpfulText="不清晰"  helpfulText="清晰"  unhelpfulId="understand\_collect\_unhelpful"  teaData={{  business\_id: '10',   scene\_id: '0iUtvojX0iaUnrSmJ\_Sfp',  platform: isFfa ? '1' : '1001',  sif\_bid: isFfa ? 1 : 1001, // 1 代表抖店，1001 代表即时零售  extra: {  ticket\_id: recordData?.ticket\_id,  rule\_id: recordData?.rule\_id,  shop\_id: (userInfo as IUserInfo)?.id || (userInfo as UserInfo)?.shop\_id,  ticket\_type: recordData?.ticket\_type  }  }} > | |

5.3.2.2 **文案数据串联**

|  |
| --- |
| 通过 4.2.1.2、4.2.1.3、4.2.1.4 的原始数据表进行加工 |

5.3.3 **文案友好配置**



5.3.3.1 **平台管理**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能点 | 功能明细 | 展示字段 | 截图 |
| 平台管理列表 | 新增平台管理列表 | * 平台 id * 平台名称 * SDK 接入方式： SIF * 编辑 |  |
| 新增平台 | 新增平台 | * 平台id （必填 用户中台 app id） * 平台名称（必填） |
| 页面权限 | 暂仅支持超级管理员操作 |  |

5.3.3.2 **业务管理**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能点 | 功能明细 | 展示字段 | 截图 |
| 业务列表 | 业务管理列表 | * 业务id * 业务名称 * 业务所属平台 * 编辑 |  |
| 新增业务 | 新增业务 | * 平台id （必填 下拉选择） * 业务名称（必填） |
| 页面权限 | 暂仅支持超级管理员操作 |  |

5.3.3.3 **场景管理**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能点 | 功能明细 | 展示字段 | 截图 |
| 场景列表 | 场景管理列表 | * 场景id * 场景名称 * 场景对应组件名称 * 组件预览 * 编辑 |  |
| 新增场景 | 指标定义/解释 | * 支持反馈 * 支持有无帮助 * 支持大模型帮助 |
| ErrorMessage | * 支持反馈 * 支持有无帮助 * 支持大模型帮助 |
| 新增场景 | 新增场景 | * 场景名称（必填） * 场景对应组件名称（必填） |
| 页面权限 | 暂仅支持超级管理员操作 |  |

5.3.3.4 **规则管理**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能点 | 功能明细 | 展示字段 | 截图 |
| 规则列表 | 规则管理列表 | * 规则id * 规则名称 * 规则适用平台 * 规则管理人 * 规则版本 * 规则状态 * 规则关联文案 * 编辑/启用/下线 |  |
| 新增场景 | 新增规则 | * 规则名称 * 规则适用平台 * 规则管理人 * 规则适用平台 * 人群圈选包 |
| 页面权限 | 暂仅支持超级管理员操作 |  |

5.3.3.5 **创建文案**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能点 | 功能明细 | 展示字段 | 截图 |
| 创建文案 | 创建文案 | * 文案展示规则（下拉，文案所属平台筛选） * 是否需要反馈 * 是否需要有无帮助 * 是否需要大模型 |  |

5.3.4 **推荐文案**

|  |
| --- |
| 通过文案评分维度重新组装 prompt 产生一个文案 |

5.3.5 **文案关联知识**

|  |
| --- |
| 抖店现有小助手能力 |

6. **进展 & 规划**

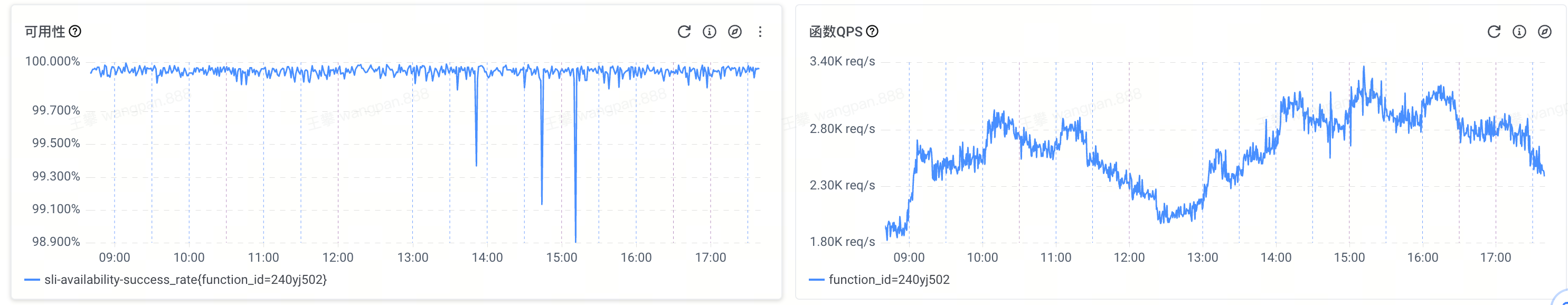
6.1 **现阶段收益**

6.1.1 **Q2 业务收益**

* 业务收益： [文案全链路监控优化](https://bytedance.larkoffice.com/docx/MZmtdFl45odkfXxKWpCcXnbXnjb)在奖惩、商品、体验平台等场景落地，收益如下：（收益为项目组整体收益，前端收益未拆分）
* 奖惩商家侧理解度上升 122%，30% -> 66.63%
* 作者理解度 49% -> 66%
* 商家奖惩 CPO：60 -> 30，下降 50%
* 通过大模型能力召回30+问题条例，推动整改
* 商品驳回相关工单咨询量：493 -> 293 下降： 40%
* 商家体验分相关指标定义咨询量：5894 -> 5333 下降 9.5%

6.1.2 **Q2 技术收益**

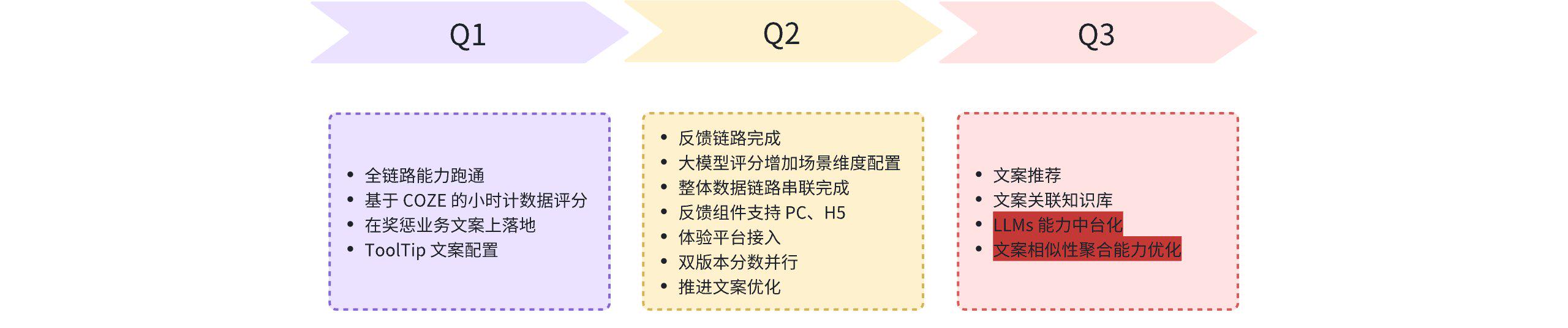
* 技术收益
* 整体基建能力完成 70%，完成消息队列、模型（gpt中台、火山方舟）、向量（火山方舟）、node 服务、前端 SDK、数据采集全链路串联；新增业务仅需发送消息到任务消费队列，配置任务、维度、样例 即可完成一个 few-shot 的模型任务上线
* 整体服务高峰期 QPS 3k+，SLA 99.9%，按目前实例运行状态 QPS 可支持到 20k。



* 模型整体成本，6月在亿级数据的量级下 模型 + 向量 库成本仅为 1900 元



6.2 **后续规划**



6.3 **评审记录**

* 文案场景考虑加入用户动线相关内容 @Smart Panwith@Jinchao Wu
* 单一维度分数很低，但是其他维度满分，阈值怎么判断

7. **参考资料**

1. [VikingDB HTTP API 使用手册](https://bytedance.larkoffice.com/wiki/wikcnslHYICgeyCI1KzcIgN0CDb?bk_entity_id=enterprise_7228450961776181249)
2. [【文案全链路监控】数据收集方案V2](https://bytedance.larkoffice.com/docx/MfNrdB22bokje1xTJYRc2q0Nnob)
3. [技术方案: LLMs 文案质量评估](https://bytedance.larkoffice.com/wiki/BfHlwvORqiZQ9fkqwPjc9Vppnzh)
4. [open api 模型接入](https://gpt.bytedance.net/gpt_openapi/model)
5. [解读: 用 LLM 进行无参考文本质量评估](https://zhuanlan.zhihu.com/p/635133696) | [原论文](https://arxiv.org/abs/2304.00723)
6. [DEE: Dual-stage Explainable Evaluation Method for Text Generation](https://arxiv.org/abs/2403.11509)
7. [ALLURE: Auditing and Improving LLM-based Evaluation of Text using Iterative In-Context-Learning](https://arxiv.org/abs/2309.13701)
8. [奖惩业务反馈数据](https://bytedance.larkoffice.com/docx/CBsidOFsGoEE3OxuV6ecpStjnMe)、[申诉业务反馈数据](https://bytedance.larkoffice.com/docx/U6I0dmQ8Wo6LCLxMnaVcMsoonNg)、[达人违规需求埋点数据](https://bytedance.larkoffice.com/docx/FVk2d3pdTo4tDBxB4iKclKganyq)
9. [产品的文案体系 – 人人都是产品经理](https://www.woshipm.com/pmd/5237973.html)
10. [一文讲清产品设计中的文案表达 – 人人都是产品经理](https://www.woshipm.com/copy/5827052.html)
11. [UI、交互、PM应该懂的产品提示文案设计\_用户](https://www.sohu.com/a/323800115_114819)