用户画像

用户活跃度

- 特征
  - Bayesian Prior:launch_life_time
  - 一阶特征:
    - avg：周期性，由远及近的活跃度消费水平变化
    - lag：last7，last14
    - cross特征：[0,1,0,1] 0->1的变化，用户活跃的稳定性
  - 二阶特征：
    - std：以上特征的有效性
  - time_since_last: 以上特征距今的时效性
  - 用户属性特征
- 目标：平均活跃概率
- 短期->长期：用户历史30天的活跃度预测分对用户进行了聚类，Hierarchy K-means Cluster
- 落地：
  - UG：短信，DSP人群促活，抖音防沉迷，数值策略

作者发文&流失

- 目标
  - Active：未来一周作者发文的概率
  - Descend： 未来一周发文概率<历史四周发文概率
- 特征处理
  - 在以上用户活跃度的基础上对作者进行了分体裁特征处理
- 落地：结合作者分层自动化作者流失召回&发文引导
  - 作者的app_age * 作者的粉丝分层 * Active * Descend
  - 站内私信触达
    - 个性化文案+号外
    - 社群热词和个性化投喂
    - 垂类活动+
- 评估：召回率，召回->发文转化->周增活跃作者->大盘VV量级，在AB实验分析中还加入了CUPED

项目整体评估

1. 优点：在于可解释，用户之间可比，可以灵活的迁移不同的下游任务
2. 问题：标签和用户策略是不完全匹配的

因果推断

数值策略

时长任务

目标：在净收入不降的前提下，最小化成本

Base数值 + [0.1,0,3,0.5]

Delta arpu>threhsold

- 最小成本组
- 次小成本组
- delta arpu最大的组

LT是不显著负向： 小部分用户是显著负向的

后期优化

- 约束优化
- 天极任务+天内衰减
- 新用户+老用户对齐
- 不同体裁引导

客户端体验

预估：不同场景体验优化对用户留存的影响上界

1. 场景
    - 启动时间
    - Feed加载时间
    - 详情页加载时间
    - FPS
    - Drop FPS：丢帧率：
    - 小视频/视频首帧
    - 手机温度
2. 背景：之前客户端主要通过劣化实验量化每个场景的影响，多数的劣化实验都不显著，业务方拿不到有效的实验结论
3. 待验证假设

    1. 同一个用户对优化和劣化的反应相同
    2. 优化和劣化相同幅度，可感知用户相同
4. CUPED：分析客户端的历史实验，解决历史劣化实验不显著
    1. 分析历史不显著的实验：都能找到局部的显著村提。且显著群体主要集中在中端机用户。中端机用户对劣化实验是最敏感的，低端机用户已经习惯了卡顿，高端机用户劣化较小的幅度是不会有明显的感知的。
    2. 结论：要达到显著的用户感知需要满足，要么用户体验绝对值下降/上升到某个阈值，或者相对下降幅度超过某个阈值
    3. 指导意义
        1. 劣化实验和HTE探索实验相同：对不同机型按设备分开不同的探索阈值。
        2. ATE得到一个二维的结果，策略可能渗透的用户占比，以及对渗透用户的平均影响，拆分新用户&老用户
5. Propensity Score：解决无法开展优化实验的问题
    1. 在每个场景找到该场景最敏感的聚合指标
    2. 直接对该指标进行分组，留存统计，得到一个粗糙的上界：confounder
    3. 通过因果推断来预估更精准的收益预估的上界
        1. 用户差异->app体验->历史活跃信息
        2. 用户差异->历史活跃信息
6. 劣化实验后的反转实验：
7. HTE: 并不是所有策略都适合进行个性化上线，成本和产品策略本身的考量。HTE进行个性化uplift的预估，然后用模型可解释方案，得到一个负向/正向人群画像，来指导业务方进行策略优化。
    1. 小视频横向滑动，+滑动劣化->留存正向。人群画像：低活+历史横向滑动max

其他分析

1. 渠道对用户留存影响分析：厂商预装>厂商商店的新用户留存显著高10个点
    1. 分析师：用用户属性对用户进行分组，发现在各个分组内
    2. applist+用户设备分
    3. applist：预装+自装app+系统APP 数上升的，Delta LT是单调下降的

NLP算法

事件匹配

LGB：我们也尝试了Sentence Bert

标注：

- 0=not match：完全无关，模版发文：体育比赛的通告，天气预报，实体/数字不同
- 相似：核心实体一致，存在部分内容增益。
- 重复 消重粒度，用户视角两篇文章没有信息增益。列表页定义，标题没有信息增益

特征：

- USE: contextualized word embedding
- BM25： weighted word embeding
- TF-IDF: 共现的关键词数
- Entity：
    - 事件构成要素: 空间LOC+时间DATE+人物PER+ORG+数值
    - 实体过于稀疏：体育&娱乐&疫情&。借鉴了小样本prototype的思路设计了Anchor向量，
    - 无实体重合：体育倾向度越高，事件不相关性越高

不使用Bert几个原因

- 需要用到content，标题完全无关但是内容相关。尤其以微博的事件
- 语义相关：类似需要实体的判断逻辑

消重&匹配

1. 粗粒度：BM25 关键词+关键词权重， word2Vec -> weighted word Embedding
2. 热榜词匹配的MUSE：样本少

热点挖掘提报

目标：及时挖掘站外热点，补充当前头条热点事件候选

- 头条已经有的事件：不要重复提报，会增加审核同学的工作量
- 头条未有的事件要及时提报，以及对当前头条还没有足够内容的站外事件，会每隔一段时间尝试进行内容召回和提报

1. 站外模块
    - data process模块：consumer-producer 接入不同站外抓取接口，kafaka，完成文本清洗，事件统一的id，生成事件格式，并写入站外榜单队列
    - feature worker：消费站外榜单队列
        - 对白名单榜单，计算事件匹配所需特征缓存到redis
        - 并把站外数据写到mysql：记录或者查询站外榜单状态，未送审，审核通过
        - 对于状态符合未送审->榜单提报队列
2. 站内榜单模块
    1. 定时轮询站内榜单，计算新增榜单特征，并把对应特征缓存到redis
3. 匹配召回模块
    1. 接入站外kafka，和当前站内榜单快快照读取redis特征，调用事件匹配服务进行匹配
    2. 未匹配榜单，调用搜索召回，写入提报队列
    3. 无召回榜单，线上审核结果，更新mysql状态
4. 全网热点构建：
    1. 时效性
    2. 覆盖率
5. 产生了哪些todo
    1. 垂类榜单覆盖率极低：科技，娱乐，财经。个性化榜单，垂类榜单提报源

2. 覆盖率漏斗：最大的折损是在可分发事件->上榜事件。热度计算公式造成线上榜单流转较慢
    1. 低ctr Deboost
    2. 事件ctr的Boost系数对高ctr的适当提升，低ctr打压。核心是在不影响热榜权威性的前提下，提高榜单流转速度，给更多候选榜单曝光机会

泛内容相关召回

背景：热点事件内流，从事件发散开的内容。最开始一路实体召回。造成召回内容相似度过高，且内容较少容易断流。

1. 改索引：
    1. 实体：过于单一
    2. 关键词：过于宽泛，无法区分有效信息和冗余信息。keywords的权重是NLP视角的，而非用户视角的
    3. 文本Embedding：细粒度消重用的多，任务目标相对明确， 样本标注简单，标准统一，准确度高
    4. UA召回不合理：
    5. Keyphraes：从用户视角衡量用户看这篇文章的时候用户是在看什么。基于搜索点击数据，每个gid，考虑不同用户对相同文章关注点不同，top1&top2的搜索query，用了细粒度分词的query terms去和标题进行匹配得到序列标注样本。
        1. match prob < threshold的样本会被过滤掉：希望尽可能多的抽取出keyphrase—>直接影响抽取数量
        2. title + title & abstract
2. 改索引的关联方式：
    1. 关联实体：Knowledge
    2. Item2Vec的思路进行建模。这里可以可以构建内容/意图相关的context，同一个搜索session的点击
    3. 构建样本时还尝试了不同的方式
        1. 一个session一个样本
        2. Ner 序列中引入query id，对相似query进行聚合
        3. 不同用户同一个query的session聚合成一个样本
        4. 一个session内，考虑点击顺序，还是不考虑顺序对所有实体进行消重得到set
3. Author2Vec：相似画风的作者召回。作者内容+作者画风
    1. 用了用户session作为样本，而不是Entity2Vec里面用Query为粒度的样本
    2. 候选召回：类小红书，类知乎画风。

内容审核

1. 质量审核的区别：可分发/适合在某个领域进行分发
    1. 独特标签
        1. 图文一致性：
        2. 疫情倾向识别
        3. 事件内召回内容一致性

2. 滑动窗口特征：给作者重新做人的机会

UC Concept Mininig

基于用户视角的Concept概念挖掘，文末tag->文章内流。

举例：

- 搜索视角
  - oppo手机：颜值高的手机，最适合拍照的手机
  - 法式餐厅：适合情侣约会的餐厅，外国人都去的餐厅
- 内容视角
  - 一条男士瑜伽裤撑起20亿的估值。文末的相关搜索给的都是瑜伽裤款式，购买这类的query。PE/VC创投，热点财经新闻，体育健康新趋势

1. keywords的索引任务，把title和相关query进行匹配
2. weak-supervised：
   1. query title alignment：会受到分词的影响，所以我们用了多个分词器之保留一致的结果
   2. bootstrap
   3. weakly supervised Bert-CRF
      1. Teacher 模型
      2. 重新初始化训练student模型拟合teacher模型的结果
3. 判别
   1. NLP 特征：
      1. 成词：左右信息熵，互信息
      2. 内容属性概率：体育，娱乐等倾向分
      3. prefix和surfix的postag类型
   2. 搜索行为特征：
      1. concept出现在多少query中，query搜索总数等 IDF
      2. concept独立作为query被搜索 TF

小组命名-改写任务

举例：

- 朱一龙->朱一龙粉丝俱乐部
- 广场舞->我们都爱广场舞
- 养生->人人都是养生家

技术方案

1. 匹配：固定的pattern，匹配每个关键词最适合的pattern
2. 改写：keyword -> 生成name。保证相关性的同时，要求生成的format要比较符合小组的格式

改写任务

1. 样本生成：知乎豆瓣爬了一波小组名。然后用keyphrase关键词抽取从小组名里抽取出关键词，剩下的就是pattern
2. 样本增强
    1. 关键词：Word Synonyms[5个不同预料训练的word2vec]，同义词库
    2. pattern：word Synonyms, MLM
    3. 关键词 * pattern组合->5K左右的样本。按pattern进行了分层采样，避免部分pattern权重过高，会发生在生成任务里全是一个pattern的预测结果
3. 模型：UNILM框架
    1. 预训练： RoSimbert
        1. Bert+UNILM+对比学习
    2. Finetune：
        1. Batch Sim
        2. 只针对小组组名的perplexity
    3. Decode： TopK Sample Decode，保证生成名称的多样性。每次返回5个命名结果让运营同学去选择

CP+

1. 样本清洗
    1. AutoEncoder来过滤Loss>threshold的
    2.
2. 特征处理：没有列归一化处理，但是我们做了行归一化处理，因为不同债券的单位可能存在差异。极大减少模型corner case出现的可能
3. Loss：Huber Loss
4. 实时的训练更新：
    1. Daily retrain的天极模型
    2. Hourly Retrain的小时模型：小时模型会用上一个hour，模型预估和实际交易价格的差异作为特征，用了类似残差学习的思路来加快模型对于市场异动的学习。在脱欧的时候我们的模型比全市场所有定价模型都更快捕捉了价格变化