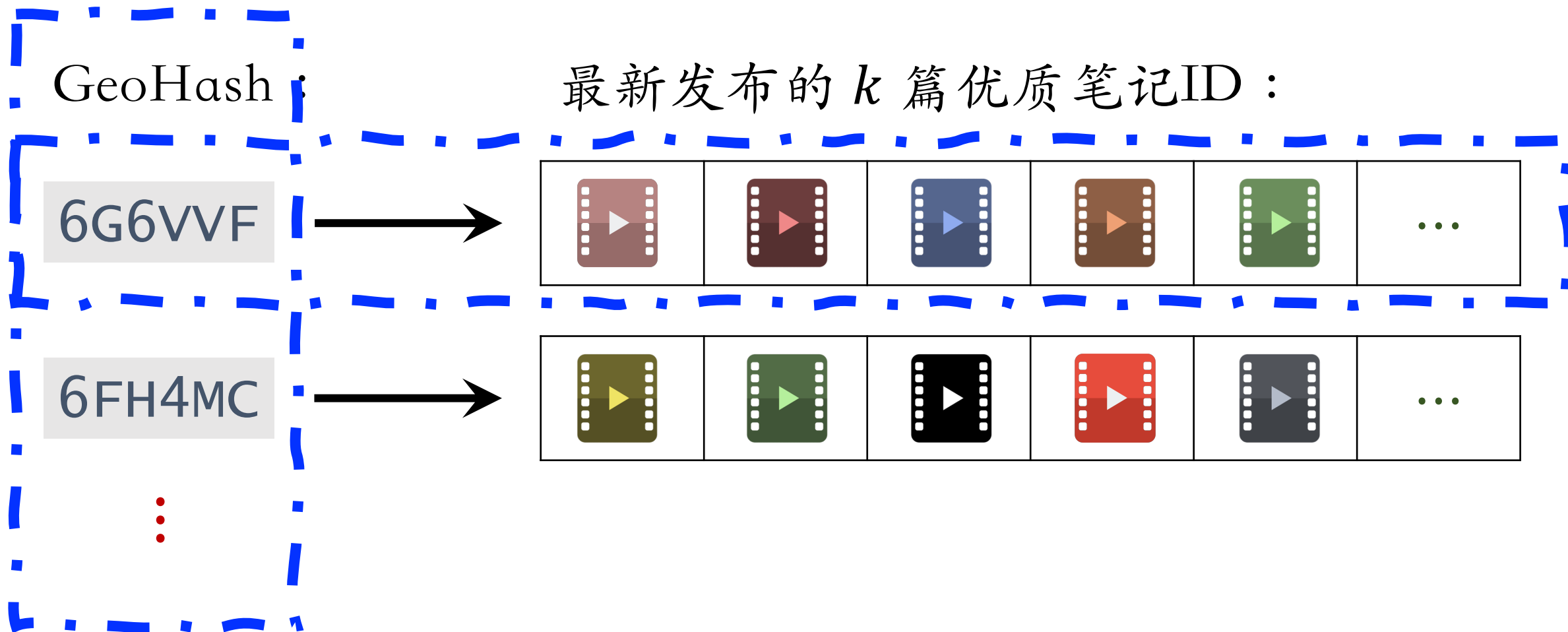


地理位置召回

GeoHash召回

- 用户可能对附近发生的事感兴趣。
- GeoHash：对经纬度的编码，地图上一个长方形区域。
- 索引：GeoHash → 优质笔记列表（按时间倒排）。
- 这条召回通道没有个性化。

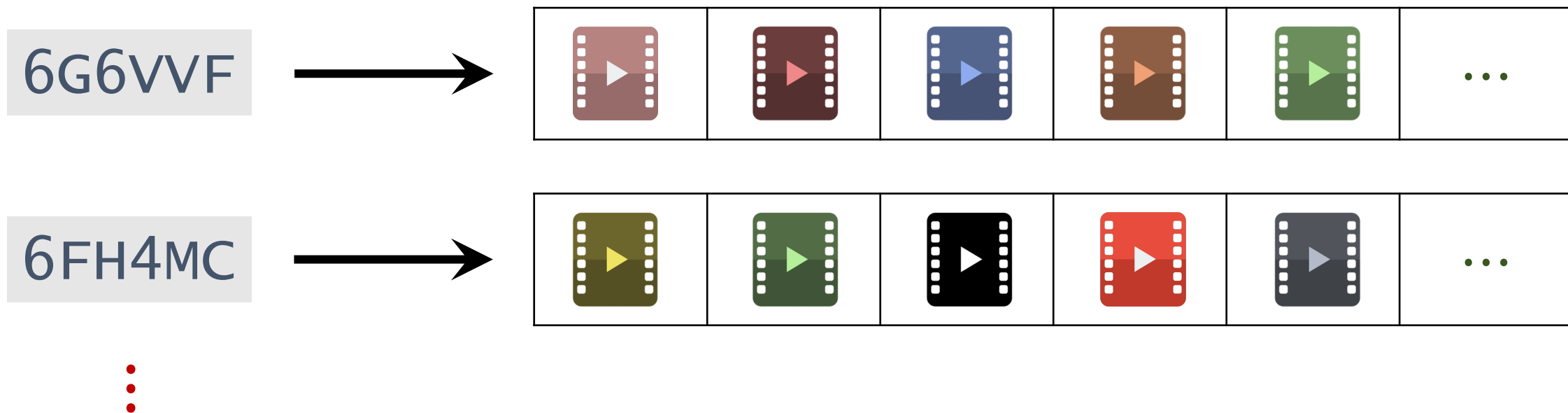
GeoHash召回



GeoHash召回

GeoHash :

最新发布的 k 篇优质笔记ID :



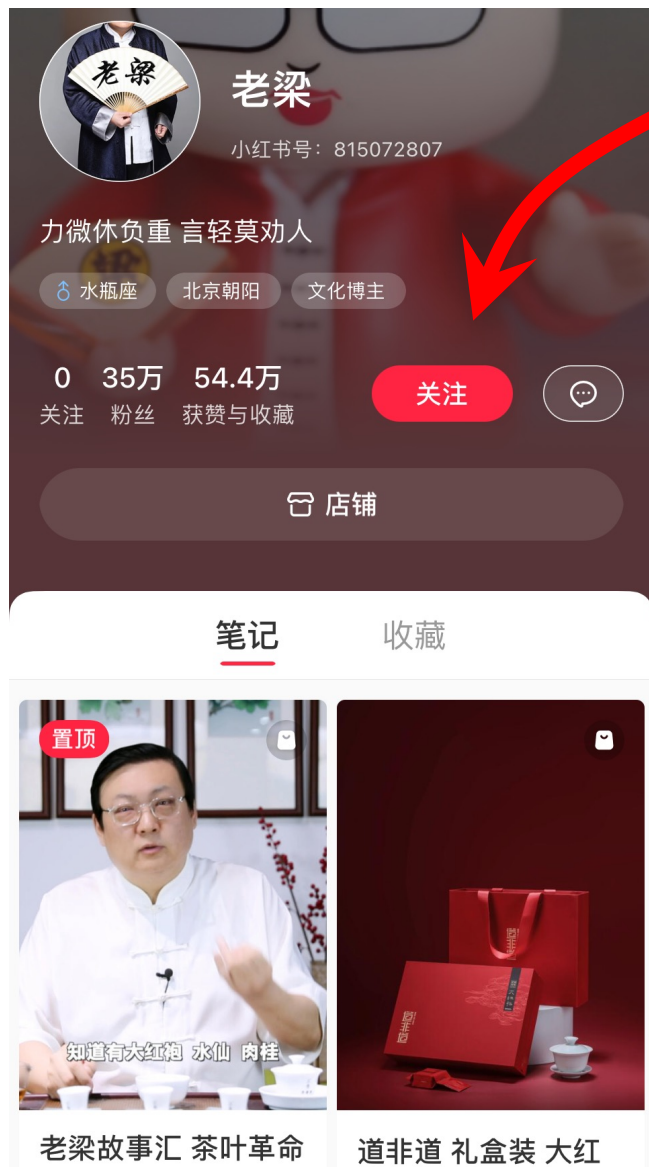
根据用户定位的GeoHash，取回该地点最新的 k 篇优质笔记。

同城召回

- 用户可能对同城发生的事感兴趣。
- 索引：城市 → 优质笔记列表（按时间倒排）。
- 这条召回通道没有个性化。

作者召回

关注作者召回



- 用户对关注的作者发布的笔记感兴趣。

- 索引：

用户 → 关注的作者

作者 → 发布的笔记

- 召回：

用户 → 关注的作者 → 最新的笔记

有交互的作者召回

- 如果用户对某笔记感兴趣（点赞、收藏、转发），那么用户可能对该作者的其他笔记感兴趣。
- 索引：用户 → 有交互的作者
- 召回：用户 → 有交互的作者 → 最新的笔记

相似作者召回

- 如果用户喜欢某作者，那么用户喜欢相似的作者。
- 索引：作者 \rightarrow 相似作者 $(k \text{ 个作者})$
- 召回：用户 \rightarrow 感兴趣的作者 $(n \text{ 个作者}) \rightarrow$ 相似作者 $(nk \text{ 个作者}) \rightarrow$ 最新的笔记 $(nk \text{ 篇笔记})$

缓存召回

缓存召回

想法：复用前 n 次推荐精排的结果。

- 背景：
 - 精排输出几百篇笔记，送入重排。
 - 重排做多样性抽样，选出几十篇。
 - 精排结果一大半没有曝光，被浪费。
- 精排前 50，但是没有曝光的，缓存起来，作为一条召回通道。

缓存召回

缓存大小固定，需要退场机制。

- 一旦笔记成功曝光，就从缓存退场。
- 如果超出缓存大小，就移除最先进入缓存的笔记。
- 笔记最多被召回 10 次，达到 10 次就退场。
- 每篇笔记最多保存 3 天，达到 3 天就退场。

总结

➡ • 地理位置召回：

- GeoHash 召回、同城召回。

➡ • 作者召回：

- 关注的作者、有交互的作者、相似的作者。

➡ • 缓存召回。