自动化运营—分类标签整理方案草案

标签体系整理流程：

选取对应的class（即对应的class\_id,class\_name）,例如取class\_id=1(即class\_name:悬疑)：

a.根据class\_id=1获取对应的书籍（对应书籍信息：book\_id,book\_name）,对应SQL语句为：select book\_id,book\_name,book\_tag,class\_id,class\_name from con\_book where class\_id=1

b.根据书籍列表汇总book\_tag信息（即该class\_id=1下，以书本为单位，统计label\_tag），如下表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| book\_name | boook\_id | book\_tag | class\_id | class\_name |
| 迷藏1：血色面具 | 5844 | 血色面具;红娘子;悬疑 | 1 | 悬疑 |
| 生死魔谷 | 5847 | 生死;破案;警察;枪声;失踪 | 1 | 悬疑 |
| 牙印 | 6581 | 牙印;故事;有鬼 | 1 | 悬疑 |
| G游记 | 6612 | G游记 | 1 | 悬疑 |
| 福尔摩斯探案精选（世界经典文学名著・青少版） | 6696 | 福尔摩斯;探案;精选;世界;经典;文学;名著;青少版; | 1 | 悬疑 |
| 福尔摩斯探案全集1 | 6697 | 侦探；法律；闻名；正义；真相 | 1 | 悬疑 |
| … | … | …… | … | … |

若统计结果如下（这里我们取前10个高频标签用于分类）：

|  |  |
| --- | --- |
| book\_tag | num |
| 悬疑 | 5678 |
| 恐怖 | 666 |
| 侦探 | 600 |
| 破案 | 567 |
| 惊险 | 555 |
| 探案 | 532 |
| 盗墓 | 517 |
| 惊悚 | 512 |
| 推理 | 505 |
| 探险 | 489 |

则我们可以选取前10的book\_tag进行分类（这里我们剔除与class\_name 同的标签：悬疑），可得分类规则如下：

盗墓探险：盗墓、探险

惊悚恐怖：恐怖、惊险、惊悚

侦探推理：侦探、推理、破案、探案

综上可得：

|  |  |  |
| --- | --- | --- |
| class\_id=1 | category\_name | category\_tag |
| 悬  疑 | 盗墓探险 | 盗墓、探险 |
| 惊悚恐怖 | 恐怖、惊险、惊悚 |
| 侦探推理 | 侦探、推理、破案、探案 |

但是按照这种分类规则，总会有一些书籍不能归属到相应的类，这种情况下我们该如何解决？

可以采取一下三种方案：

方案一：另以class\_name命名一个类，例如此类我们可命名为：悬疑\*\*，这里我们暂且命名为—悬疑怪谈。

方案二：余下的书籍我们利用book\_brief，首先，判定book\_brief中是否含有category\_tag，若有，则划定到相对category中，若没有，则根据book\_brief，给予打上合适类的关键标签。

注：如果可以，我们应当book\_tag，进行处理，保证其为2-3个汉字为宜！

要实现以上标签整理方案：我们需要先建立两张表：

ag\_id与tag\_name 关系表；book\_id（book\_name）与tag\_id（tag\_name）的对应表

最后我们的目的是得到以下两张表：

book\_id 、tag\_id、count(tag\_id)三者的关联表；class\_id、category\_id、category\_tag三者的关联表

自动化运营—分类标签整理方案草案（补充）

为了提高图书画像和用户画像的准确性，我们需要对分类标签库，进行标准化的整理和扩展。

大体可分以下几步：

第一步：对现有的标签库进行整理，剔除无价值标签。

第二步：扩展标签库，根据现有的图书信息重新给图书打标签。这里我们采用分词的方式进行，具体如下(这里我们采用Python作为开发语言)：

(1)我们所拥有的图书信息：图书名（name），图书大类（class\_name），图书长短简介(breif)，图书章节名(charpter\_name)，图书内容(content)。这里我们本选取这五个信息进行分析，但基于图书内容过大，不仅会使很大程度上扩大运算量，还会应分词过多过杂乱，从而干扰到分词打标签的效果。鉴于此我们只取前四个内容，作为分词打标签的基础信息。

(2)我们采用Python中的大结巴（jieba模块）对基础信息进行分词，然后用Python中的counter模块，对分词进行聘书统计和排序整理。详见附录一代码。

(3)构造和完善停用词库、标签库、新词库。这里我们称之为图书基因工程的项目，标签词则是项目中的基础单位：基因元。停用词库，顾名思义就是分词中的不可作为标签的词语的集合，也就是图书基因工程项目中的非基因元库。标签库，即可作为图书标签的词语库，也就是图书基因工程项目中的基因元库。新词库，即还不确定是停用词，还是是标签词的新词语库。这里我们先以通用的通用词库作为基础停用词库，然后通过跑程序不断的训练，完善停用词库，并构建自己的标签库。

(4)根据停用词库和标签词库，对图书重新打上标签。对新词库，根据新书的新增情况，和训练样本产生的量级，进行定期处理即可。

第三步：构建图书画像和用户画像，实现自动运营。

(1)通过标签体系给给各大类图书汇总高频标签，从而产生书单和子类，实现分类和书单的自动化运营（关于自动运营-分类和书单，下面会详细介绍）

(2)通过图书标签给构建图书画像和用户画像，实现个性化推荐的自动化运营。（关于自动化运营-个性化推荐，下面会详细介绍）