周工作总结2019/04/15-2019/04/19

本周完成了房评审核这边的几个新规则开发和badcase修复，以及线上全量评论的simhash值分布分析，同时完成了看点这边的文本去广告迭代，增加了14条新关键词，并完成了测试集上的测试，提高了覆盖率，误伤率也降低了一个点。本周实践最多的是写正则了。

技术积累

1. **倒排索引**

   倒排索引（英语：Inverted index），也常被称为**反向索引**、**置入档案**或**反向档案**，是一种索引方法，被用来存储在全文搜索下某个单词在一个文档或者一组文档中的存储位置的映射。它是文档检索系统中最常用的数据结构。

   一般的索引是文档——>关键词的模式，而倒排索引则相反，你拿到一个关键词，就能立马找到它在哪个文档里出现过，这样再把那个文档揪出来，效率高。

   这个是在做房评去重的准备工作时，崔鸣给安排的，分析出每个关键词对应的候选队列的长度，就能知道如何建立整个的倒排索引了。

2. **正则基础——贪婪与非贪婪**

   这个知识点起源于一个我没有理解的地方，就是我写的正则

   ```
   (?:[0-9一二三四五六七八九两]?[十百千万]?)+(?:[0-9一二三四五六七八九两]?)
   ```

   这段正则在匹配"好地3435段低总价的小复式楼中楼三房一厅78万"的时候，不会直接匹配到结果，反而得到['', '', '3435', '', '', '', '', '', '', '', '', '', '', '', '三', '', '一', '', '78万', '']类似于这样的结果。但是如果把两后面的问号改成+号就可以了。

   ```
   (?:[0-9一二三四五六七八九两]+[十百千万]?)+(?:[0-9一二三四五六七八九两]?)
   ```

   结果是['3435', '三', '一', '78万']。

   在研究这个问题的时候发现，如果re.compile("(xx)|(xx)|(xx)|(xx)")就会得到四个组，当然跟解这个问题没关系哈。然后把正则简化到'[0-9一二三四五六七八九两]*[十百千万]?'还是会匹配到一堆""，甚至'[0-9]*'还是很多个""，每一个匹配不到的字符，都返回一个""。这个问题先留着吧，后面再研究研究。

3. **hive从json字段提取内容**

```
hive -e
"select project,  task_id,attribute_key , cardContext from
dw.dw_ser_sinan_task_biz_attribute_da
LATERAL VIEW json_tuple (attribute_value,'cardContext') v1 AS
cardContext where pt = '20190409000000' and project=
'newHouseReview' and attribute_key= 'auditContext'"
```

这一段的意思就是从attribute_value里取了cardContext字段的值并取名为cardContext，最后select这个字段。 后由于cardContext里面有”\n”,造成一行内容会分行，所以用regexp_replace(cardContext,”\n”,””)替换掉换行符就可以了。

4. **excel计算频率的函数frequency**

   excel的frequency函数，要注意的地方是写公式的时候要先选中要输出到的表格位置（这个位置要跟分割条件一一对应），然后输入”=frequency(S1:S78, W1:W7)”,S的位置是候选数据，W的位置是分割条件。

异常问题

1. **正则出现死循环**

   本周在写正则的时候遇到很长时间出不来结果的正则，看起来像是陷入了死循环，但最终也能跑出结果，具体的case是

   ```
   '78.456.346.45.47576.43'
   ```

   正则是

   ```
   (?:[0-9一二三四五六七八九两]+\.*\d*[十百千万]?)+[0-9一二三四五六七八九两]?(?:每平米|每平方|每平|每米|/㎡|每㎡)
   ```

   后来经过分析，认为前半部分的[0-9]会和\d*有歧义，所以会计算很长时间，把\d*去掉就可以了。

个人反思

1. 这周在写正则的时候发现对于原理还是了解的太少了，所以很多正则写出来自己都不知道会有什么结果，也不知道对不对，只能验证一下，所以后面应该专门找个时间来整理一些资料和实践的例子出来。