了解推特与api原数据（1.0版本）

编辑人：杨杰

协助者：赵海林

一：需要了解的内容与概念：

1、Snowflake 用于产生推特ID，这个是基于时间的，所以对于ID，它是不断向上增长的。（https://dev.twitter.com/overview/api/twitter-ids-json-and-snowflake）

2、推文：指的是用户发布的一句话，如下图所示。



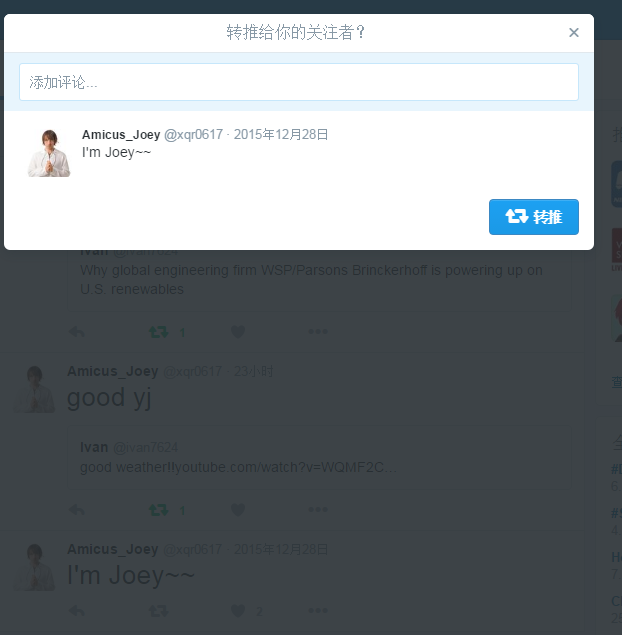
上图中，joker06是他的name，kaqyugosann是他的screen\_name（唯一标识），2016年3月7日是发布时间（这个需要搞清楚，这里的时间，在推特后台存储的是UTC标准时间，如果你用的电脑是北京时间，则在网页上它会自动处理成与之对应的北京时间，所以如果你的电脑如果用东京或则纽约时间，这个显示的发布时间可能有所差别，但是我们储存的是UTC时间，这个全世界都一样。再则，如果你将鼠标放在推特显示的时间上时，还会出现一个时间，这个时间与你登陆账号设置的时区有关，如果你设置的美国时区，这将会少16个小时）。“emoji test”是作者发布的一句话，即一条推文。下面的符号从左往右，依次是，回复、转推、对推文点赞、推广（需要付钱）、更多。

3、回复

点击上图空白部分，你可以看见其他推特用户对这条推文的回复，当然点击回复，你也可以回复这条推文。

4、转推（retweet）

点击上图第二个键，即可转推这条推文。如下图所示。



如果你什么评论也不加，直接点击转推，则为转推推文。转发推文原则上是一条推文。

5、引用推文（quoted）

如果上图你加了评论再转推这条推文，则为引用推文。引用推文原则上是新的一条推文。

6、原始推文截断：

使用api采集数据时，当text文本字段大于140时，会出现截断现象。

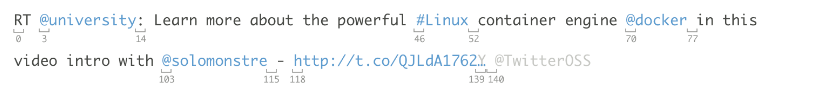
二、FAQ：

问题一：为什么推特用户和推文等既有int64，又有id\_str？

答：主要是像Javascript等不支持大于53-bits。

问题二：当推文是转推时（即retweet），为什么最好分析retweeted\_status的数据？

答：首先必须知道当这条推文为转推时，retweeted\_status这个字段必然存在，这是同时存在的，判断这条推文是不是转推，不是用retweeted字段（这是说明这条推文有没有被采集API用户转推没有），而是用retweeted\_status这个字段判断有无。其次，举个例子：



如上图所示，这是一条转推推文，原文的数据是少于140个字节的，因此没有被截断，但是变为转发推文后增加了14个字节，导致原文截断，而trancated字段却显示没被截断（false）。这会导致解析text出现错误，最直接的就是，在外层entities字段中@用户提及只会出现3个人，不会出现@TwitterOSS。这时，如果解析retweeted\_status可以得到更为完整的信息。

三、Rest\_api与streaming\_api的差异：

（注：测试采用的rest\_api中的statuses/user\_timeline和search/tweets，streaming\_api中的statuses/filter）

1、原始数据中Rest\_api有而streaming\_api没有的字段：

Metadata：每个search/tweets上都有一个，字典型，包括两个字段 iso\_language\_code和result\_type,如{‘iso\_language\_code’：en,’ result\_type’:recent}但是感觉没啥用，毕竟这两个参数都可以用其他参数来确定，可以不管。但是statuses/user\_timeline中没有这个字段。

2、原始数据中Rest\_api没有而streaming\_api有的字段：

1）filter\_level:有3种水平，none(默认)，low,medium,（将来可能有high），官方解释说none可以返回所有的有价值的推文，所以会出现有些推文与关键字无关的情况，一般在一个会议上演示用medium就行了，medium基本上与在twitter.com上搜索的结果相当。

2）timestamp\_ms:毫秒级的UTC（13位）。

3） extended\_tweet:包括full\_text，entities，extended\_entities ，display\_text\_range(这个指的是仅仅推文文本的长度)

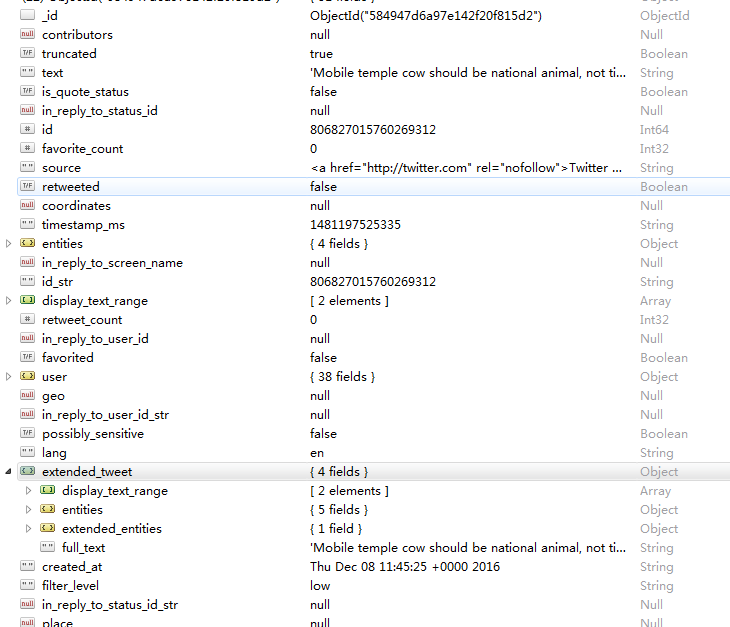
4) display\_text\_range。

2、关于字段的具体解释（不同的情况，不同的api采集的原始数据有所不同！！）：

2.1.在statuses/filter中

2.1.1关于截断字段（truncated）

当truncated为true时，会出现extended\_tweet字段，这里面包含了full\_text等信息。（rest\_api至少我没发现这个字段）,如下图所示。Quoted\_status和retweeted\_status字段也有extended\_tweet字段。



2.1.2 关于is\_quote\_status字段：

当推文是引用时，有两种情况，第一种是直接引用，即在别人的推文上加上自己的见解，这时会出现is\_quote\_status为True，quoted\_status也会出现（如果引用的被删除了就没了，见下图）。



2.1.3 关于retweeted\_status字段：

当推文是转推时，会出现retweeted\_status这个字段，如果遇见这种情况（见下图）：推文B引用了推文A，推文C转发了推文B，那么推文C的原始推文将会有retweeted\_status和quoted\_status这两个字段，并且is\_quote\_status是为True，而且retweeted\_status中也包含了quoted\_status字段，（我找了20万组推文发现这两个quoted\_status内容是一样的，但是赵海林发现了不一样的，当我们想去研究时已经找不到了，这个问题留给后人确定吧）。与引用一样，如果这种情况下引用被删除了，那么两个quoted\_status都消失了，不过，is\_quote\_status还是为True。

（在这里再次强调并提个建议，当推文是转推时，最好解析retweeted\_status字段的内容，最上层的text中，可能被截断，导致最上层的entities解析不完整。）

2.1.4 关于推文文本（text）后面的链接。

Text里面的url链接，都是<http://t.co/>... ，不知道你们看了字段解释表（Entities表）没有，在对推文文本解释中，出现了url，media和extended\_entities这三个字段，你可能对这三个字段有所疑惑，我来解释一下。首先我们应该需要明白推文的text是希望将所有的信息都包括进去的，那么如果你上传了一张图片，推文怎么显示？好了，正如entities表中解释的一样media用于显示图片信息（现在media字段只能显示图片，不知道以后官方会不会扩展），media中url是推特压缩过的链接（即<http://t.co/>。。。）,去访问这个链接将会跳转到expanded\_url这个完整的链接，media\_url则是下载图片的链接，sizes中包含了4种图片形式（很小，小，中（默认），大），主要是看有没有被压缩（crop），当然media指的是推特上面的图片，不一定是自己上传的。。。。如果你发布推文时，加了一个其他网站或者是其他推文链接（也有可能是因为text太长，截断后，推特帮你加上的），那么，url这个字段将会显示完整信息。有时你可能会看见text文本中有两个链接，但是有时url字段有两个解析信息，有时只有一个，有时甚至没有！不要奇怪，你这时会发现media中的信息个数加上url的信息个数会等于text文本中的链接数。讲到这里你应该明白哪些链接代表推特上传图片，那些代表不是推特上传图片的链接了吧。

如上图所示，引用推文中有一条youtube的链接，quoted\_status字段的text中一定会有一个<http://t.co/>...链接，而且entities中会有url字段，不会有media字段。现在你可能又有了新的问题，media只能上传图片，视频怎么办，还有大于一张图片怎么办？好了，extended\_entities字段中video\_info可以存储视频链接，有两张图片时，extended\_entities将会有两个信息。

注：这里强调一下，解析streaming\_api时，最好解析extended\_tweet和extended\_entities，因为有可能text中的链接指向自己这条推文，那么自己上传的图片可能不会包含在text文本和entities字段中，但是在extended\_tweet和extended\_entities中会出现。如下图所示：

第一张图是推文的网页显示，第二张图中，可以看见text被截断了。但是解析后的url中的expanded\_url为<https://twitter.com/i/web/status/tweet_id（tweet_id>是自己这条推文的ID），这说明指向了自己，而图片信息并没有显示。第三张图是这条推文的extended\_tweet在这里面有着media的信息。



图1 网页显示

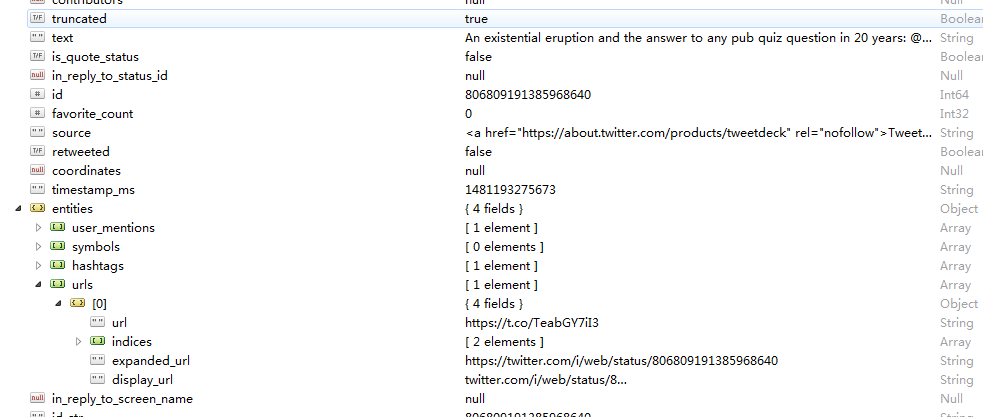


图2 推文的原始字段

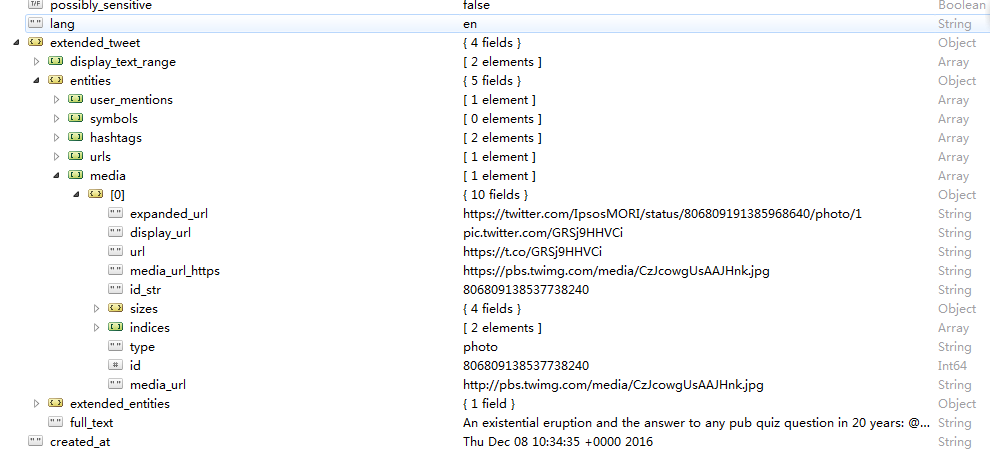


图3 推文的extended\_tweet字段

Statuses/filter 这个api就介绍到这里，找点原始数据与网页上的推文去对照吧。

2.2.在rest\_api中的user\_timeline中：

2.2.1 关于truncated字段：

区别于streaming\_api，因为没有extended\_tweet这个字段，因此truncated如果为true，那么我们无法从原始数据中获取完整的推文，想要获取完整推文，只有另想办法。因为所有的twitter\_api都一样，如果text字段被截断了，那么推特会在text后面自动加上一个<http://t.co/>...，这个链接你可以在entities字段中找到，它的expanded\_url为<https://twitter.com/i/web/status/tweet_id，直接使用requests.get>这个地址，我们可以得到完整的text，当然这个方法，可以访问任何推文ID，从而获取回复信息，如果有需要，再进行研究。有时如果truncated为False，但是这条推文是转推推文，那么text也有可能截断（理由上面已经讲过），这时最好就去分析retweeted\_status字段。

注：rest\_api都大同小异，具体的差异由于时间有限也就没有研究了，如果需要使用请自行研究。

四、实例：

讲了这么多，来个例子：

下面3个图分别是一条推文的所有字段、quoted\_status字段、entities字段，问题1、它是用哪种API采集的原始数据？2、它是哪种类型的推文（原始、转推、直接引用、转推引用）3、这条推文是否被截断，如果截断了，它的完整信息在哪里？4、这条推文的发布者是否只发布了文本信息，有没有其他链接信息，如果有它是哪种信息？5、如果原推文中去掉引用，下面的图片数据会有哪些变化？

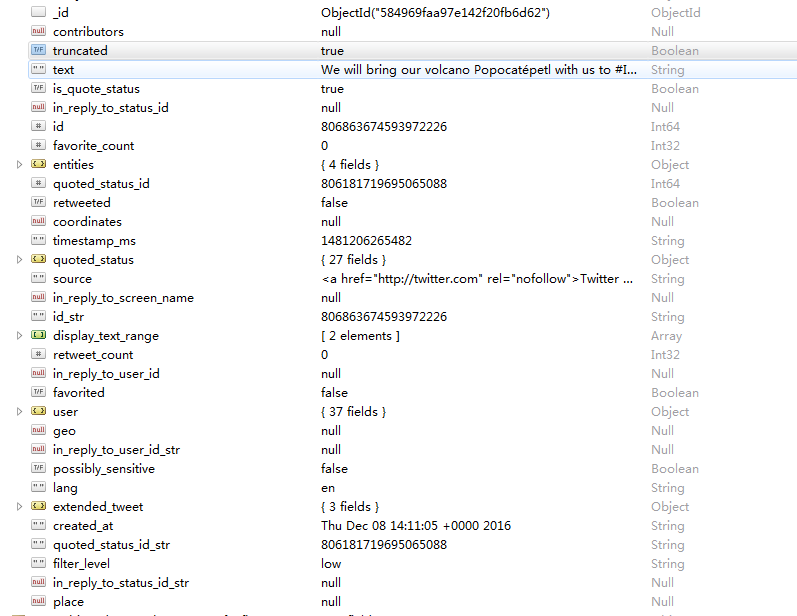


图1一条推文的所有字段

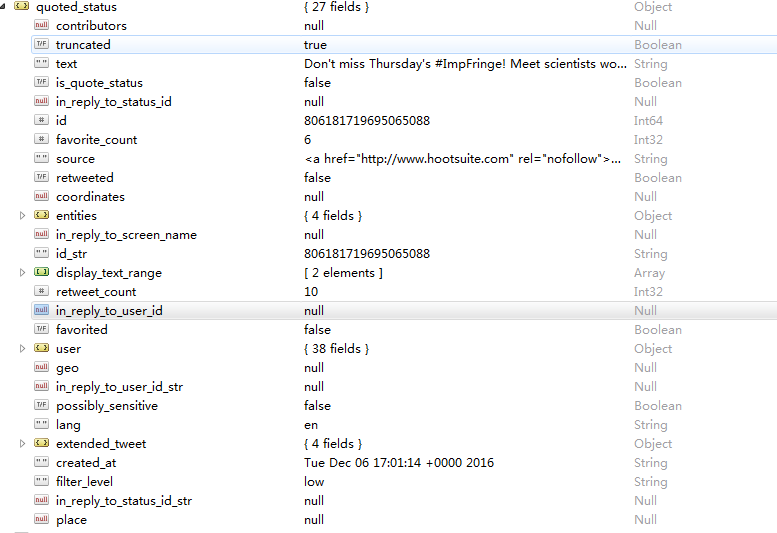


图2推文的quoted\_status字段



图3 推文的entities字段

答案：1）streaming\_api中的filter 2）直接引用 3）截断了，完整信息在extended\_tweet中 4）有链接，这是由于截断了，推特自己生成了一个指向自己推文的链接。5）quoted\_status字段会消失。