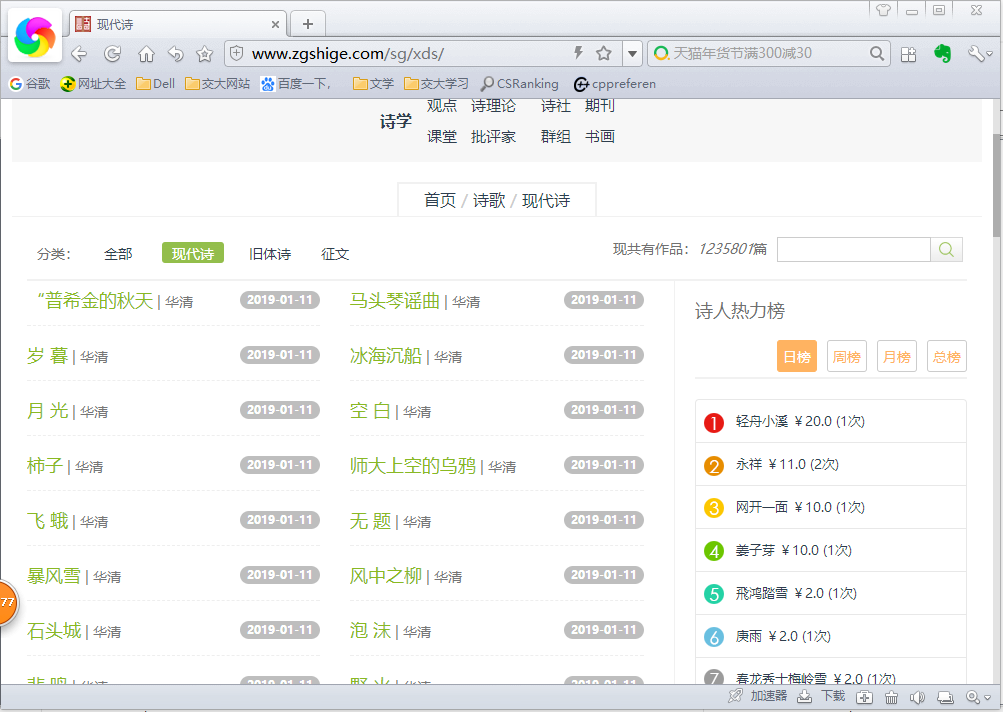
爬虫、网站信息提取

环境：python2.7 ,urllib2, beautifulsoup4

第一部分：中国诗歌网上诗歌数据获取

网页来源：中国诗歌网的现代诗、旧体诗部分

爬取数量：10k+



诗歌大全网站中现代诗目录



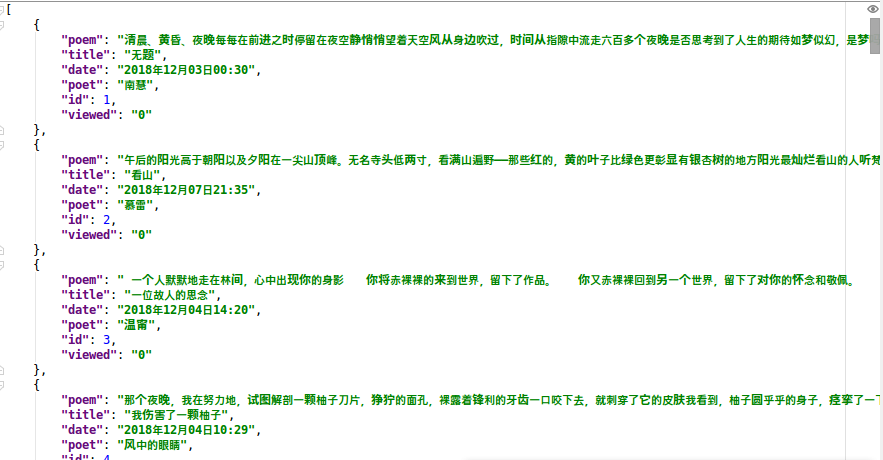
中国诗歌网现代诗页面

这里的网站没有反爬虫机制，可以进行比较方便的爬取。网站目录中一页有60个现代诗的网址。注意到网站信息包含诗的ID信息体现在网站的URL中，但ID是随机跳跃的递增序列，所以我们无法直接根据URL的信息来获取所有网站。

我们一开始采取的方法是先选取一个现代诗的页面作为种子，然后进入该页面根据其“下一页”的链接作为爬取的下一个网站。但这样的做法有一个很大的缺陷是爬虫的鲁棒性太低，因为一个页面最多只能生成下一个页面的链接，遇到错误时程序就会终止。

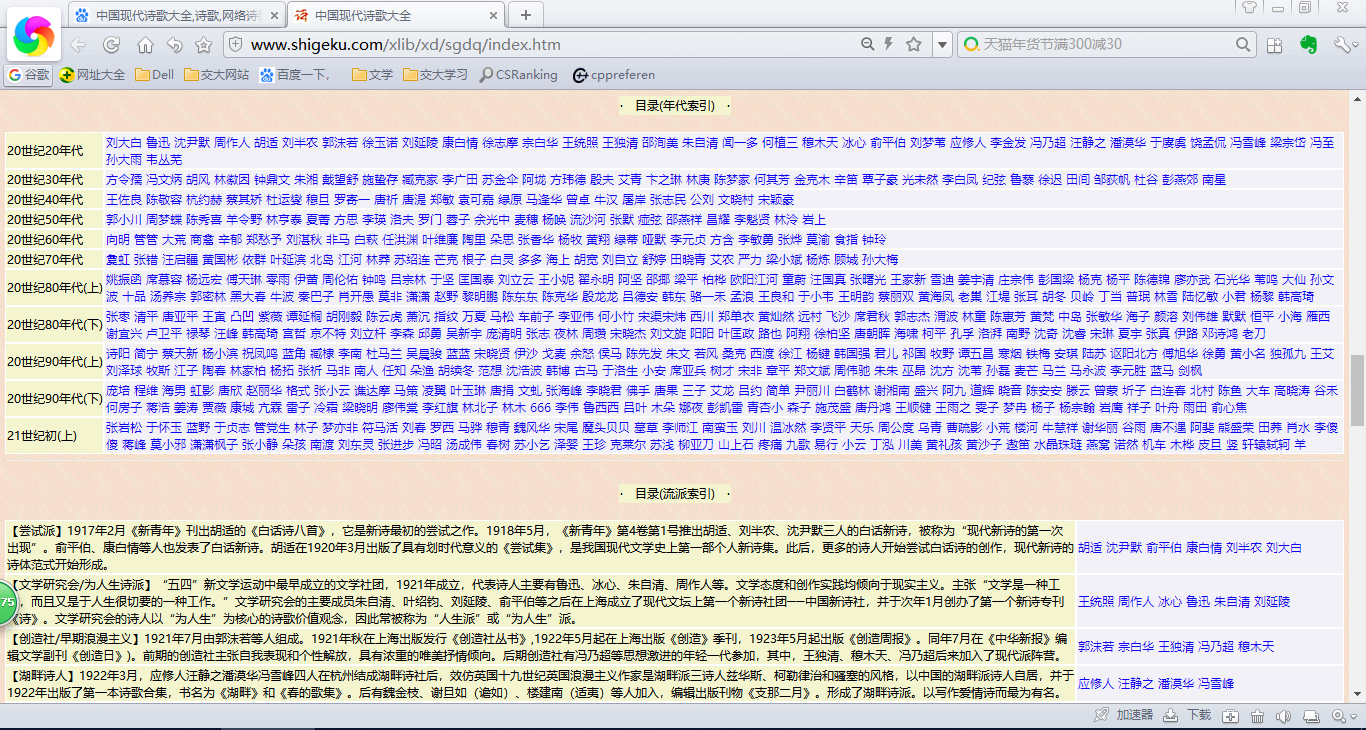
我们最后的解决方法是根据目录中的页数，每10页选择1个种子，并使用多线程进行爬取，每次爬去依然选择使用用“下一页”获取下一个链接放入队列，并设置每次爬取的最大页面数为5000。由于队列先进先出，我们对队列依次出队的操作可以大致保证每个初始种子所引出的页面数大致相当，这样我们一次爬取的5000个网站中便能保证其中的页面不会有重复。

我们使用正则表达式匹配文档标签，并且输出成json格式。



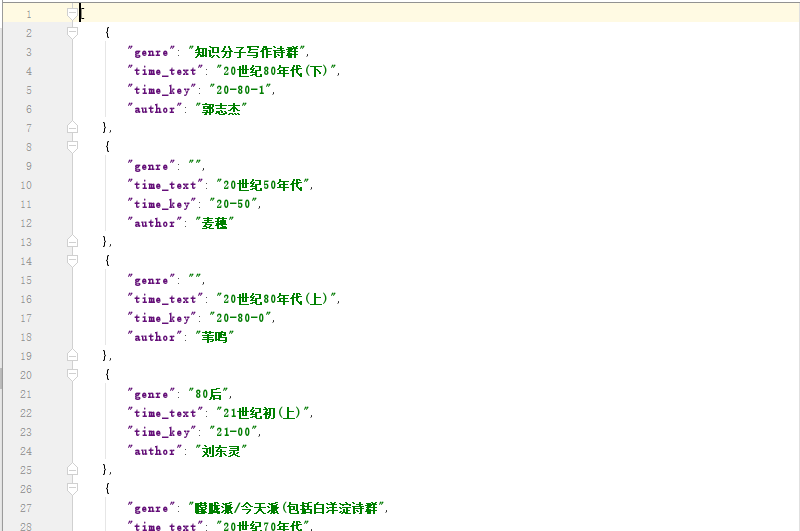
result1-40.json数据

第二部分：根据诗歌大全网站整合作者年代、流派信息



诗歌大全作者目录

这里我们用beautiflsoup解析网页内容，并且从中交叉提取同一个作者的年代和流派等信息，输出json格式。



cnmodern\_author\_info.json

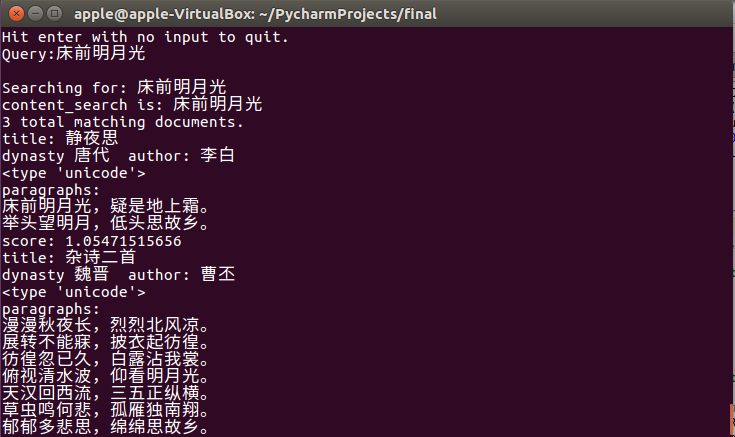
索引操作

环境：lucene4.9.0，Elastic Search、Kibana 6.5.4

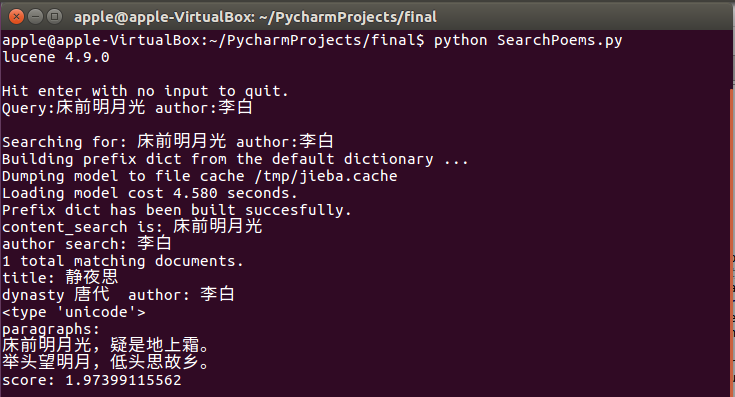
我们初始用lucene建立索引，而后根据我们更大量数据与更深层次的搜索，我们放弃lucene，而选用Elastic Search。

第一部分 lucene的尝试

lucene的索引建立是基于结巴分词，将内容、标题（以及关于古诗文的赏析、译文）进行分词，并将其和诗歌的其他信息一起，根据需要选择是否存储与建立倒排索引。由此对应的索引搜索将诗歌的内容作为必须匹配项，而将标题、作者、朝代信息作为“或”逻辑匹配项进行组合搜索，由此可以提升作者想要搜索内容在搜索结果页面的次序。



lucene文本搜索



lucene组合搜索

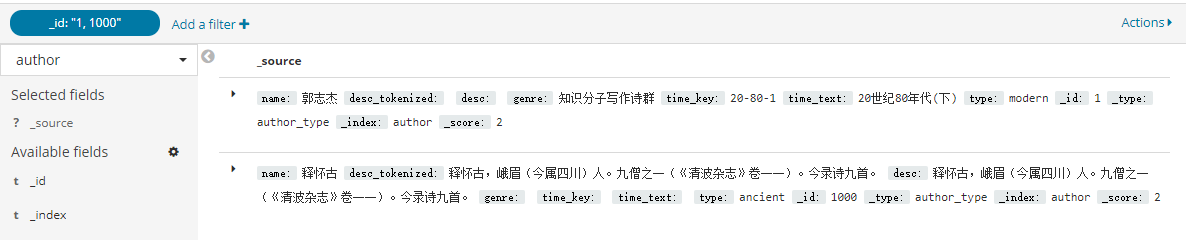
1. Elastic Search更新索引

在索引建立完成之后根据我们之前重新获取的作者信息以及生成的新标签，我们需要对索引对原有索引进行更新。更新主要包括：添加现代作者的信息（流派、年代），整合古现代作者的索引，添加诗歌标签等。

关于添加现代作者流派、年代信息，我们先将所有作者添加三个空项 ”time\_text”, ”time\_key” ,”genre”分别表示年代原文（20世纪30年代），年代关键词（20-30），流派（“七月派”），然后再根据我们获取的数据将json转换为字典，然后运用update\_by\_query()方式，对作者姓名做匹配，并添加相应信息。

关于整合古现代作者，我们将古代、现代作者合并到一个”author”索引中，并新增一个”type”类型来区分古代、现代作者（分别用“ancient”,”modern”区分）。

关于添加诗歌标签，我们主要运用jieba.analyse.extract\_tags()方法。对于现代诗歌，因为其已经与我们日常用语很接近，所以我们直接用结巴的默认库，对其提取最相关联的5个标签。我们限定我们提取的标签为名词(“n”)以及地名(”ns”)，因为这些词语一般能包含更加具体和重要的信息。对于古代诗歌，我们仍然是基于tf-idf算法进行关键词抽取，用set\_dictionary()的方法将主词典设置为古词词典(“guci\_dict.txt”)，并设置频率文件: jieba.analyse.TFIDF(idf\_path='guci\_idf.txt')，频率文件为在我们数据库里的诗歌中所有词语所出现的频率，我们运用jieba.analyse.extract\_tags()方法选取前三个匹配，并且添加到原来古诗”label”的后面（原来的”label”是基于诗歌的分类，比如“唐诗”、“给孩子的诗”这类）。最后，我们用update\_by\_query()的方法，按诗歌ID对古诗和现代诗分别添加和修改”label”项。



古、现代作者索引整合