python使用esmre代替ahocorasick实现ac自动机[多模匹配]
(http://xiaorui.cc/2015/06/13/python%e4%bd%bf%e7%94%a8esmre%e4%bb%a3%e6%9b%bfahocorasick%e5%ae%9e%e7%8e%b0ac%

6-13 views

为什么会用AC自动机？ 如果你想知道一篇文章有没有你要过滤的敏感词，怎么办？ 不可能用正则一个个的匹配吧？ 敏感词超过300个之后，用Trie来构建模式树 (字典树)的速度优势相当的明显... ...

Hello Boys，文章的原文转自 http://xiaorui.cc (http://xiaorui.cc) http://xiaorui.cc/?p=1649 (http://xiaorui.cc/?p=1649)

特别说下，trie图也是一种DFA，可以由trie树为基础构造出来，对于插入的每个模式串，其插入过程中使用的最后一个节点都作为DFA的一个终止节点。

如果要求一个母串包含哪些模式串，以用母串作为DFA的输入，在DFA 上行走，走到终止节点，就意味着匹配了相应的模式串。

ps: AC自动机是Trie的一种实现，也就是说AC自动机是构造Trie图的DFA的一种方法。还有别的构造DFA的方法...

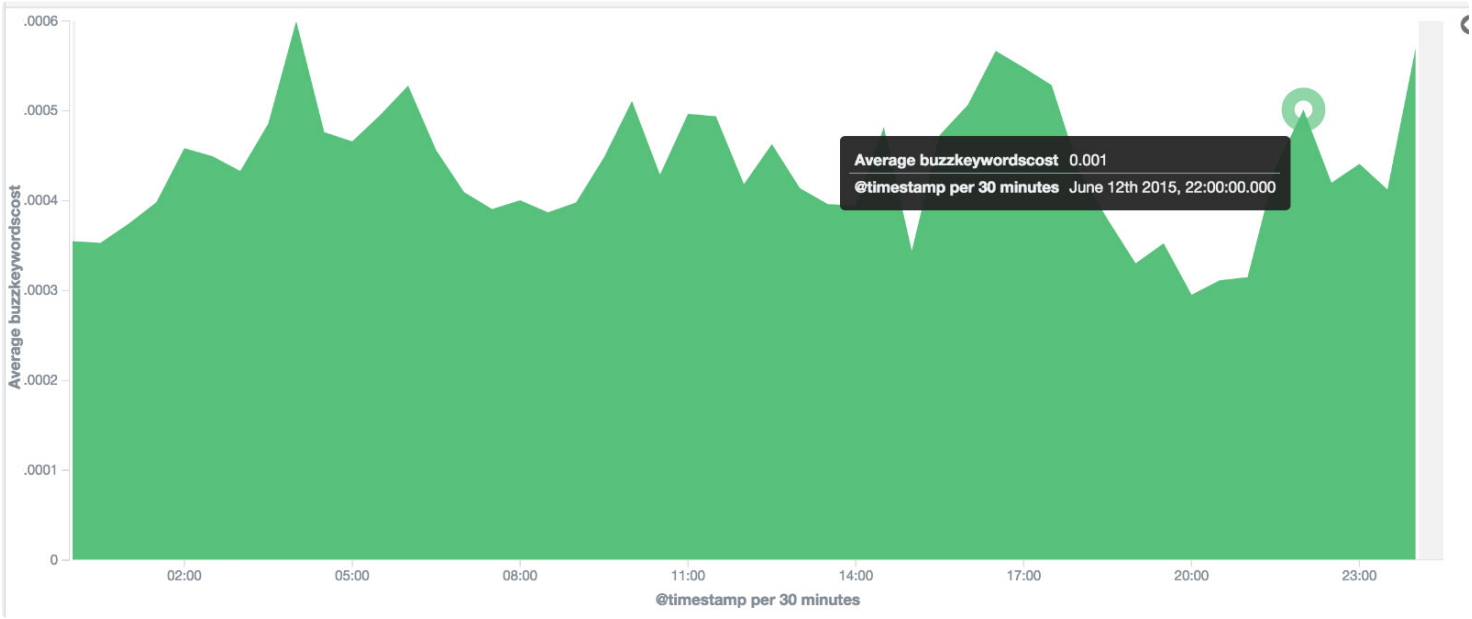
不扯淡了，我们后端都是python写的，python的ahocorasick模块跟我们的业务不太匹配，问题是这样的 ！ 如果你的服务是用来做敏感词匹配，也就是说所有文章里面只要含有一个关键词，那就说明匹配了。 但是我们的业务是文章中的所有能匹配到的关键词都一一的抽取出来。 我想有些朋友可能还不太明白，那么我举个例子， 如果我的关键词里面有宝马和马，那么用python的ahocorasick库只会得到宝马，而不会得到马。 问题是处在马这个字节是在宝马的链条里面的。 如何避开这个问题，我们也懒得自己重写了，已经有人给出了解决的模块。 已经测试完成，并上线使用了。

安装简单，直接pip install esmre

```python
import esm
index = esm.Index()
index.enter("宝马")
index.enter("马")
index.enter("奔驰")
index.enter("保时捷")
index.fix()
index.query("哎呀，今天在楼下看到了宝马，我老家倒是有养马的，以前的邻居有个奔驰，不对是保时捷，大爷的，都是马")
```

```
In [1]: import esm

In [2]: index = esm.Index()

In [3]: index.enter("宝马")

In [4]: index.enter("马")

In [5]: index.enter("奔驰")

In [6]: index.enter("保时捷")

In [7]: index.fix()

In [8]: index.query("哎呀，今天在楼下看到了宝马，我老家倒是有养马的，以前的邻居有个奔驰，不对是保时捷，大爷的，都是马")
Out[8]:
[((33, 39), '\xe5\xae\x9d\xe9\xa9\xac'),
 ((36, 39), '\xe9\xa9\xac'),
 ((63, 66), '\xe9\xa9\xac'),
 ((93, 99), '\xe5\xa5\x94\xe9\xa9\xb0'),
 ((111, 120), '\xe4\xbf\x9d\xe6\x97\xb6\xe6\x8d\xb7'),
 ((141, 144), '\xe9\xa9\xac')]
```

测了几天，性能还是可以的，500KB的文章，6000多个关键词，消耗的时间在0.002左右，相比ahocorasick一点都不差的。 观察了下，esmre是没有发现内存异常泄露等问题。



Python

```
[2015-06-12 23:34:01,043] INFO extractor "Get keywords takes 0.0003 seconds"
[2015-06-12 23:34:01,069] INFO extractor "Get keywords takes 0.0002 seconds"
[2015-06-12 23:34:01,178] INFO extractor "Get keywords takes 0.0002 seconds"
[2015-06-12 23:34:02,372] INFO extractor "Get keywords takes 0.0002 seconds"
[2015-06-12 23:34:02,386] INFO extractor "Get keywords takes 0.0012 seconds"
[2015-06-12 23:34:02,631] INFO extractor "Get keywords takes 0.0002 seconds"
[2015-06-12 23:34:03,656] INFO extractor "Get keywords takes 0.0021 seconds"
[2015-06-12 23:34:03,744] INFO extractor "Get keywords takes 0.0001 seconds"
[2015-06-12 23:34:03,785] INFO extractor "Get keywords takes 0.0001 seconds"
[2015-06-12 23:34:03,910] INFO extractor "Get keywords takes 0.0002 seconds"
[2015-06-12 23:34:04,031] INFO extractor "Get keywords takes 0.0002 seconds"
[2015-06-12 23:34:05,004] INFO extractor "Get keywords takes 0.0035 seconds"
[2015-06-12 23:34:05,579] INFO extractor "Get keywords takes 0.0055 seconds"
[2015-06-12 23:34:05,602] INFO extractor "Get keywords takes 0.0005 seconds"
[2015-06-12 23:34:05,662] INFO extractor "Get keywords takes 0.0010 seconds"
[2015-06-12 23:34:06,125] INFO extractor "Get keywords takes 0.0002 seconds"
[2015-06-12 23:34:06,299] INFO extractor "Get keywords takes 0.0002 seconds"
[2015-06-12 23:34:06,404] INFO extractor "Get keywords takes 0.0003 seconds"
[2015-06-12 23:34:07,396] INFO extractor "Get keywords takes 0.0002 seconds"
[2015-06-12 23:34:07,595] INFO extractor "Get keywords takes 0.0004 seconds"
[2015-06-12 23:34:08,725] INFO extractor "Get keywords takes 0.0015 seconds"
[2015-06-12 23:34:09,504] INFO extractor "Get keywords takes 0.0004 seconds"
[2015-06-12 23:34:09,515] INFO extractor "Get keywords takes 0.0005 seconds"
[2015-06-12 23:34:10,650] INFO extractor "Get keywords takes 0.0002 seconds"
[2015-06-12 23:34:11,206] INFO extractor "Get keywords takes 0.0003 seconds"
[2015-06-12 23:34:12,298] INFO extractor "Get keywords takes 0.0002 seconds"
[2015-06-12 23:34:12,319] INFO extractor "Get keywords takes 0.0001 seconds"
[2015-06-12 23:34:13,547] INFO extractor "Get keywords takes 0.0006 seconds"
[2015-06-12 23:34:13,853] INFO extractor "Get keywords takes 0.0005 seconds"
```









(http://xiaorui.cc/2014/09/21/python%e4%b8%8b%e7%9a%84ahocorasick%e5%ae%9e%e7%8e%b0%e5%bf%ab%e9%80%9f%e7%9a%84%e5%85%b3%e9%94%ae%e5%ad%97%e5%8c%b9%e9%85%8d/)
python下的ahocorasick实现快速的关键字匹配

(http://xiaorui.cc/2014/09/21/%e5%88%86%e4%ba%abpython%e5%ae%9e%e7%8e%b0%e7%9a%84lfu%e7%bc%93%e5%ad%98%e6%a8%a1%e5%9d%97-pylfu/)
分享python实现的lfu缓存模块-pylfu

(http://xiaorui.cc/2014/09/09/python%e8%b0%83%e7%94%a8zookeeper%e7%ae%a1%e7%90%86thrift%e9%9b%86%e7%be%a4%e7%9a%84%e5%8f%af%e7%94%a8%e6%80%a7%e5%ae%9e%e7%8e%b0/)
python调用zookeeper管理thrift集群可用性

(http://xiaorui.cc/2014/11/09/%e5%88%a9%e7%94%a8google-authenticator%e6%9e%84%e5%bb%ba%e5%b9%b3%e5%8f%b0%e7%9a%84otp%e5%8a%a8%e6%80%81%e5%af%86%e7%a0%81%e7%b3%bb%e7%bb%9f/)
利用google authenticator构建平台的otp动态密码系统

(http://xiaorui.cc/2014/11/09/%e6%b5%85%e8%ae%a8twisted%e4%b8%ad%e5%bc%82%e6%ad%a5%e8%b0%83%e7%94%a8%e7%9a%84%e6%96%b9%e5%bc%8f%e5%8f%8a%e7%ba%bf%e7%a8%8b%e7%9a%84%e5%ba%94%e7%94%a8/)
浅谈twisted应用中异步回调的方式及线程的应用

(http://xiaorui.cc/2014/11/09/%e8%ae%a8%e8%ae%baredis%e7%9a%84hyperloglog%e5%ae%9e%e7%8e%b0%e5%9f%ba%e6%95%b0%e7%bb%9f%e8%ae%a1%e6%80%a7%e8%83%bd%e6%b5%8b%e8%af%95/)
讨论redis的HyperLogLog做基数统计性能测试

(http://xiaorui.cc/2014/11/09/python%e4%bd%bf%e7%94%a8disque%e5%ae%9e%e7%8e%b0%e6%93%8d%e4%bd%9cdisque%e6%b6%88%e6%81%af%e9%98%9f%e5%88%97/)
python使用pydisque实现操作disque消息队列

(http://xiaorui.cc/2014/11/09/python%e4%bd%bf%e7%94%a8round-robin%e8%bd%ae%e8%af%a2%e6%8e%a5%e5%8f%a3/)
python使用round-robin)轮询接口

(http://xiaorui.cc/2015/05/05/%e6%96%b0%e6%a8%a1%e5%9d%97ngx_stream_core_module/)
nginx1.9新模块ngx_stream_core_module



(http://xiaorui.cc/2014/09/01/python%e5%ae%9e%e7%8e%b0%e6%9c%80%e5%b0%91%e4%bd%bf%e7%94%a8%e7%ae%97%e6%b3%95lru%e5%8c%85%e6%8b%ac%e5%ad%97%e5%85%b8%e5%92%8clist%e9%98%9f%e5%88%97/)
python实现最少使用算法lru包括dict和list队列

(http://xiaorui.cc/2014/09/04/%e5%9f%ba%e4%ba%8e%e6%9c%80%e5%b0%91%e4%bd%bf%e7%94%a8%e9%a2%91%e6%ac%a1%e7%9a%84lfu%e7%bc%93%e5%ad%98%e6%b7%98%e6%b1%b0%e7%ae%97%e6%b3%95/)
基于最少使用频次的LFU缓存淘汰算法

(http://xiaorui.cc/2014/09/01/python%e4%bd%bf%e7%94%a8gipc%e5%ae%9e%e7%8e%b0gevent%e7%9a%84%e5%a4%9a%e8%bf%9b%e7%a8%8b%e5%8f%8aipc%e9%80%9a%e4%bf%a1/)
python使用gipc实现gevent的多进程及ipc通信

上一篇
(http://xiaorui.cc/2015/06/07/%e5%a6%82%e4%bd%95%e7%94%a8python%e7%ae%80%e5%8d%95%e7%9a%84%e8%ae%be%e8%ae%a1%e5%bc%80%e5%8f%91%e5%bc%82%e6%ad%a5%)

下一篇
(http://xiaorui.cc/2015/06/16/%e4%bd%bf%e7%94%a8python%e5%ae%9e%e7%8e%b0linux%e6%9c%8d%e5%8a%a1%e7%ab%af%e7%9a%84prefork%e6%a8%a1%e5%bc%8f/)

别倒贴，要不整死你

原文地址：http://xiaorui.cc/2015/06/13/python%e4%bd%bf%e7%94%a8esmre%e4%bb%a3%e6%9b%bfahocorasick%e5%ae%9e%e7%8e%b0ac%e8%87%aa%e5%8a%a8%e6%9c%ba%e5%a4%9a
(http://xiaorui.cc/2015/06/13/python%e4%bd%bf%e7%94%a8esmre%e4%bb%a3%e6%9b%bfahocorasick%e5%ae%9e%e7%8e%b0ac%e8%87%aa%e5%8a%a8%e6%9c%ba%e5%a4%9a%e6

转载时必须以链接形式注明原始出处及本声明。