**

TextRunner (iepy) 开放式的知识图谱构建工具

**
先吐槽一下，官方的安装介绍对于汉化真是大坑啊

- 先来看下官方文档苛刻的安装条件 http://iepy.readthedocs.io/en/latest/installation.html

  > python 3.4以上
  > jdk 1.7其他版本为经过测试
  > ubuntu 14.04 其他版本不稳定

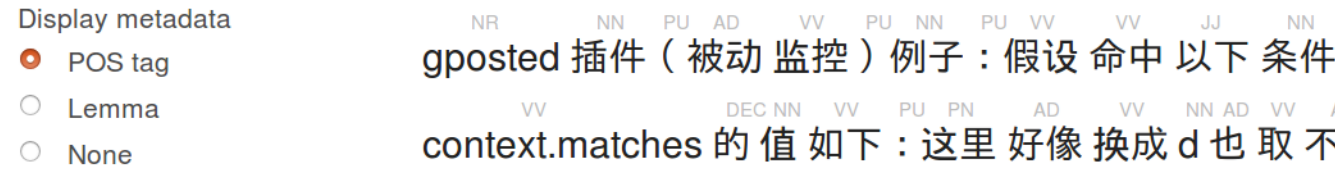
- 在看下起依赖的cornlp的汉化依赖

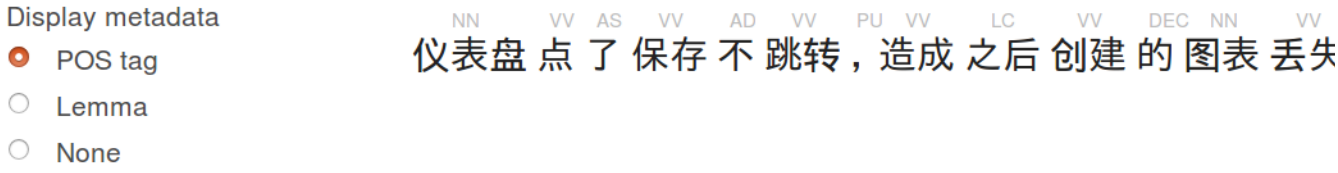
  > corn在支持中文版本的环境中java必须8以上
  > https://stanfordnlp.github.io/CoreNLP/index.html#download

  这里不细看看文档就会出现大坑啊，中文的language zh 真的不认识，会报各种错误

- 解决办法及汉化过程的参考文献
  1、为了让iepy支持高版本java，在其源码中修改stanford的高阶版本，并修改相应的配置
  将其从3.3.1 升级到了3.9.1
  iepy下载地址：https://github.com/machinalis/iepy.git
  cornlp下载地址：https://stanfordnlp.github.io/CoreNLP/index.html#download
  （这里要根据词性标注、命名实体识别语法树等下载不同的分布式包）
  2、为了支持中文zh操作
  在源端添加支持，并按照https://kkbac.wordpress.com/2016/11/25/从命令行使用斯坦福-corenlp/ 配置中文命令行启动环境

至此汉化落地工作已全部完成，效果如下：





可以看到分词、词性都没有什么问题。

那就真的没有问题了吗？？？我们继续执行
1、导入数据：这里将全部的运维数据整理 的格式导入到数据库中

```
python bin/csv_to_iepy.py data.csv
```

2、将数据进行处理分词、词性标注

```
 python bin/preprocess.py
```

到这里我们通过web浏览看到数据全部成功了，启动web

```
python bin/manage.py runserver
```

3、实体的抽取
iepy中提供了两种方式进行实体的抽取

- Running the active learning core：简单来讲就是根据定义的ENTITY来抽取三元组，当初还以为可以根据POS定义自己想要的实体，结果是不可行的：
  CORNNLP可以识别八种定义好的实体，比其他的工具种类更多：

  ```
  Location
  Person
  ```

```
Organization
Date
Number
Time
Money
Percent
```

但我们可进行的操作仅限为这八种实体，那么可不可以根据POS来进行标注呢？答案不可以
首先我们定义了一些关系词：

为了方便测试，我找了运维数据之外的一些含有实体的新闻。运维数据中很少能出现这样的实体
这些实体关系能抽取出写候选实体进行标注，下面换用POS属性，

NN为名词标记，该三元组理应抽取出关系词两端的名词来，但不幸的是并没有
为什么呢？在其源码中我找到了这样的描述：

```
class Relation(BaseModel):
    name = models.CharField(max_length=CHAR_MAX_LENGHT)
    #must be a named-entity
    left_entity_kind = models.ForeignKey('EntityKind', related_name='left_relations')
    right_entity_kind = models.ForeignKey('EntityKind', related_name='right_relations
```

在这中方式中两个词必须是命名实体，在我们的数据上行不通。

- Running the rule based core：就是按照规则来抽取指定的三元组，对不同的关系人工的设定不同的规则
  Demo文件: 抽取出生关系的规则
  https://github.com/machinalis/iepy/blob/develop/examples/birthdate/was_born_rules_sample.py
  里面就是根据词性设置的限制条件来抽取三元组

```python
@rule(True)
def born_date_and_death_in_parenthesis(Subject, Object):
    """
    Ex: Carl Bridgewater (January 2, 1965 - September 19, 1978) was shot dead
    """
    anything = Star(Any())
    return Subject + Pos("-LRB-") + Object + Token("-") + anything + Pos("-RRB-") + anyt
```

那我们来尝试下，发现没有数据没有相关的tutorial,,我把举例子中的全部数据那了下来实验给定的rule，不幸的是。。。

```
Loading candidate evidence from database...
Getting labels from DB
Sorting labels them by evidence
Labels conflict solving

Matches for rule 'born_date_and_death_in_parenthesis' (value: True)
-------------------------------------------------------------------
  nothing matched


Matches for rule 'born_date_and_place_in_parenthesis' (value: True)
-------------------------------------------------------------------
  nothing matched


Matches for rule 'born_date_in_parenthesis' (value: True)
---------------------------------------------------------
  nothing matched


Matches for rule 'born_two_dates_in_parenthesis' (value: True)
--------------------------------------------------------------
  nothing matched
```

竟然没有一条规则是成功的，什么原因呢？找啊找
终于在其文档中发现了唯一的一棵语法树，和我们目前运行出来的结果有些差距啊

```
(ROOT
  (S
    (S
      (VP (VBN Join)
        (NP (DT the) (JJ dark) (NN side))))
    (, ,)
    (NP (PRP we))
    (VP (VBP have)
      (NP (NNS cookies)))))
```

```
[27/Jun/2018 08:30:55] "GET /static/js/vendor/mod
[27/Jun/2018 08:30:58] "GET /admin/ HTTP/1.1" 200
[27/Jun/2018 08:30:58] "GET /admin/corpus/iedocum
[27/Jun/2018 08:30:59] "GET /admin/jsi18n/ HTTP/1
(ROOT
  (S
    (S (VP (VB Join) (NP (DT the) (JJ dark) (NN s
    (, ,)
    (NP (PRP we))
    (VP (VBP have) (NP (NNS cookies)))))
[27/Jun/2018 08:31:10] "GET /corpus/navigate_docu
[27/Jun/2018 08:31:10] "GET /static/css/segment.c
[27/Jun/2018 08:31:10] "GET /static/css/document.
```

左边是例子，右边是真实结果

> 到这里可以发现，其实第二种方法和我之前实现的三元组抽取方式是一样的：规则的罗列
> 但作为一个框架应该功能不止于此吧，认为下面才是重点

4、model的训练分析
加入以上步骤全部成功了，我们会得到大量的三元组，那么哪个三元组才能真正代表了句子的真实意思呢？？
三元组的选择成为关键，iepy中将其作为一个分类模型来搞。将这些数据进行人工标注，根据标注结果进行分类。
根据的特征如下:

```
number_of_tokens
symbols_in_between
in_same_sentence
verbs_count
verbs_count_in_between
total_number_of_entities
other_entities_in_between
entity_distance
entity_order
bag_of_wordpos_bigrams_in_between
bag_of_wordpos_in_between
bag_of_word_bigrams_in_between
bag_of_pos_in_between
bag_of_words_in_between
bag_of_wordpos_bigrams
bag_of_wordpos
bag_of_word_bigrams
bag_of_pos
bag_of_words
```

可以使用的模型如下：

```
    sgd: Stochastic Gradient Descent
    knn: Nearest Neighbors
    svc : C-Support Vector Classification
    randomforest: Random Forest
    adaboost: AdaBoost
```

针对每个关系设计一个分类器计算最终结果

**

总结

**
1、这是一个不错的开源框架，GUI做的很丰富；
2、两种不同的三元组抽取方式都有集成；
3、在候选三元组的过滤上将其当成一个分类问题来搞（可借鉴）