Drawline 说明文档

林基远

June 17, 2013

Contents

1 简介

Drawline 是一个**基于规则的关系抽取组件** 。它对MTIE (Mengtao's information Extraction) 进行重写，目的是更好地维护以及减少抽取时间。目前，它支持:

- **定义概念**
  - CONCEPT:concept_name:sequence_of_words 如CONCEPT:COUNTRY:中国
  - REGEX:UPSTR:[A-Z]{2,}
  - MCONCEPT:mconcept_name:(word|CONCEPT|REGEX) 如
    MONCEPT:SCHOOL:LOCNUM小学
    CONCEPT:LOC:北京
    CONCEPT:NUM:第[1-9]
    由于定义了LOC，NUM，*drawline* 会自动将LOCNUM拆成LOC和NUM，不需在它们之间加空格（否则认为空格也需匹配）。*drawline* 拆分单词时优先考虑长单词，如还定义了一个
    CONCEPT:LOCNUM:北京第一
    drawline 就不会拆分LOCNUM，而是将LOCNUM当作一个整体。
- **定义规则** ，如

  MCONCEPT_RULE:obj_pattern(per, loc, time, org):
  (SENT, "_per{PERSON}", "_loc{LOCATION}", "_time{TIME}", "_org{ORG}")

其中PERSON, LOCATION, TIME和ORG是定义的CONCEPT或MCONCEPT类型的概念。而per, loc, time, org是参数名，将会在结果输出。**模式规则可以任意嵌套使用**, 如

MCONCEPT_RULE:obj_pattern(per, loc, time, org):
(SENT, (DIST_2, "_per{PERSON}", "_loc{LOCATION}"), (OR, "_time{TIME}", "_org{ORG}"))

目前支持的操作符有：

- **AND**: 所有子句都出现的字符串才会被匹配
- **OR**: 只要有一个子句出现，该字符串就会被匹配
- **SENT**: 所有子句都出现在同一个句子，该字符串才会被匹配
  - *drawline* 以".?!。？！"及换行符来断句
  - 换行符是为了处理如表格型等无明确句号的文章
- **ORD**: 所有子句按规则定义的顺序同时出现，该字符串才会被匹配
- **DIST_n**: 所有子句同时出现在字符串，且相邻子句实例距离不超过n 个词时，该字符串会被匹配。*drawline* 将词定义为：
  - 单个中文符号（包括汉字，中文标点符号）
  - 断句符号（如上述）
  - 英文单词
  - 数字串
  - 连续的英文标点符号及空格

2 使用及结果说明

2.1 编译

- 编译成可执行的组件，在当前目录输入

```
$ make
```

- 编译成库

```
$ make lib
```

2.2 运行

编译后*drawline* 将会在当前目录产生，使用方法如下，其中config是配置的模板文件，text是要抽取的文本文件，out为结果文件。

```
$ ./drawline config text >out 2>/dev/null
```

2.3 结果说明

- 默认结果格式如下。在*****上的是抽出的关系，*****下的是概念。
  对于每个关系，第一行为offset和len，即关系对应字符串在原文本中的byet-offset，byte-len。接下来是关系，冒号前是规则的参数名，冒号后是实体实例。接着以一行——结束该关系。

```
34440 21
action:建立
person:薛建军
-----
*****
NAME:吕尧臣
NAME:徐汉棠
```

- 另一种格式如下，要生成这种格式，需要在编译时去掉Makefile里关于
-DDRAWLINE_BEAUTY_OUTPUT
的注释。方括号内数字如1580(1606)表示第1580个单词，第1606个宽字符。

```
1 facts:

-----------------
FACT 0: [1580(1606)-1593(1618)] /NAME_NAME_COEXIST/: 周志良】【周志和】【周国芳
3 ARGS:
ARG 0 [person] : 周志良
ARG 1 [person] : 周国芳
ARG 2 [coexist] : 和

-----------------

1214 concepts:
CONCEPT 0: [1(1)-3(2)] /LOCATION/: 宜兴
CONCEPT 1: [21(22)-23(23)] /ACTION/: 欢迎
```

2.4 注意事项

- 模板及文本文件都需要是UTF-8编码，结果文件也是UTF-8编码
- *drawline* 对输入规则的先后顺序没有要求
- 定义概念及规则，严格按照简介的说明定义，如不要用中文冒号引号
- 如果不需匹配，匹配串不要出现多余空格（特别是首尾），如
CONCEPT:NAME: xxx
这里xxx前的空格会被当作需要匹配的空格。
- 如果满足条件的字符串在文本多个位置都可能出现，结果文件会把所有位置都输出
- 同一名字可以有不同规则，同一名字的规则会被认为是同一类型的
- 对于一个字符串同时满足多条规则的，*drawline* 的处理为：
  - 如果这多条规则是同一类型（名字相同），则只输出一次。如定义了
  MCONCEPT_RULE:NAME_AGE(person, age):(SENT, "_person{NAME}", "_age{AGE}")))
  MCONCEPT_RULE:NAME_AGE(person, age):(DIST_5, "_person{NAME}", "_age{AGE}"))
  CONCEPT:NAME:A
  CONCEPT:AGE:[1-9] year
  这两条规则会匹配"A is 5 years' old."，但只输出一次。
  - 如果这多条规则是不同类型，每种类型都会输出一次。如上面将第2个NAME_AGE修改为NAME_AGE2。

- MCONCEPT_RULE内的子句对于以下划线开头的变量认为是参数，如“_person{NAME}”，如果要在这个子句加入其他概念，概念名不要包含下划线。若概念名必须有下划线的话，就拆成两个子句，如

子句	*yes* or *no*
“TITLE_person{NAME}”	*yes*
“OTHER_TITLE_person{NAME}”	*no*
“OTHER_TITLE”, “_person{NAME}”	*yes*

3 调用及修改

3.1 调用库

主要步骤为将规则push进*drawline* ，然后匹配即可：

```
Drawline::push(const std::string &one_rule);
Drawline::match(const std::string &text);
```

具体请参考drawline_driver.cpp。

3.2 程序原理

以下是程序主要流程及原理，需求变动时可根据下面选择模块修改。

- 输入规则转换
  在push完规则进行match前，*drawline* 会先对规则进行预处理，转化为一个分层图，层次与规则嵌套深度有关，即
  - CONCEPT，REEGX为叶子节点，在0层
  - MCONCEPT层次为它子句中最大层次+1，如
    MCONCEPT:NAME:FN LN
    MCONCEPT:FN:NN
    REGEX:N:[A-Z]?[a-z]+
    CONCEPT:LN:Jobs
    则LN, N为0层（叶子）节点，FN为第1层的节点，NAME为第2层的节点
  - MCONCEPT_RULE层次为它子句中最大层次+1，子句会被当成临时规则保存，如
    MCONCEPT_RULE:NAME_BIRTHDAY(person, birthday):(ORD,(SENT,(DIST_20, “_person{NAME}”, “BIRTH_OR”, “_birthday{DATE}”)))
    产生的规则及层次为

编号	规则	层次
1	_person{NAME}	1
2	BIRTH_OR	0
3	_birthday{DATE}	1
4	(DIST_20, . . .)	2
5	(SENT, (. . .))	3
6	(ORD, (. . .))	4
7	NAME_BIRTHDAY	5

其中1，3，4，5，6是中间节点。这部分代码主要在

```
Drawline::transform_rules()
```

- 叶子节点匹配
  - 对CONCEPT节点建立Aho-Corasick自动机，并匹配
    这部分代码在AhoCorasick命名空间里。
  - 使用boost::xpressive进行REGEX匹配
    这部分代码在

    ```
    Drawline::lowlevel_match()
    ```

- 由叶子节点构造满足规则的字符串
  从左到右，当扫到第k个叶子节点时，按照分层图，看这个叶子节点是否触发了规则。有的话构造包含这个叶子节点的父节点，再看父节点能否触发规则，能的话继续递归构造下去。
  这部分代码主要在

  ```
  Drawline::highlevel_match()
  Drawline::up_construct()
  Drawline::left_construct()
  ```