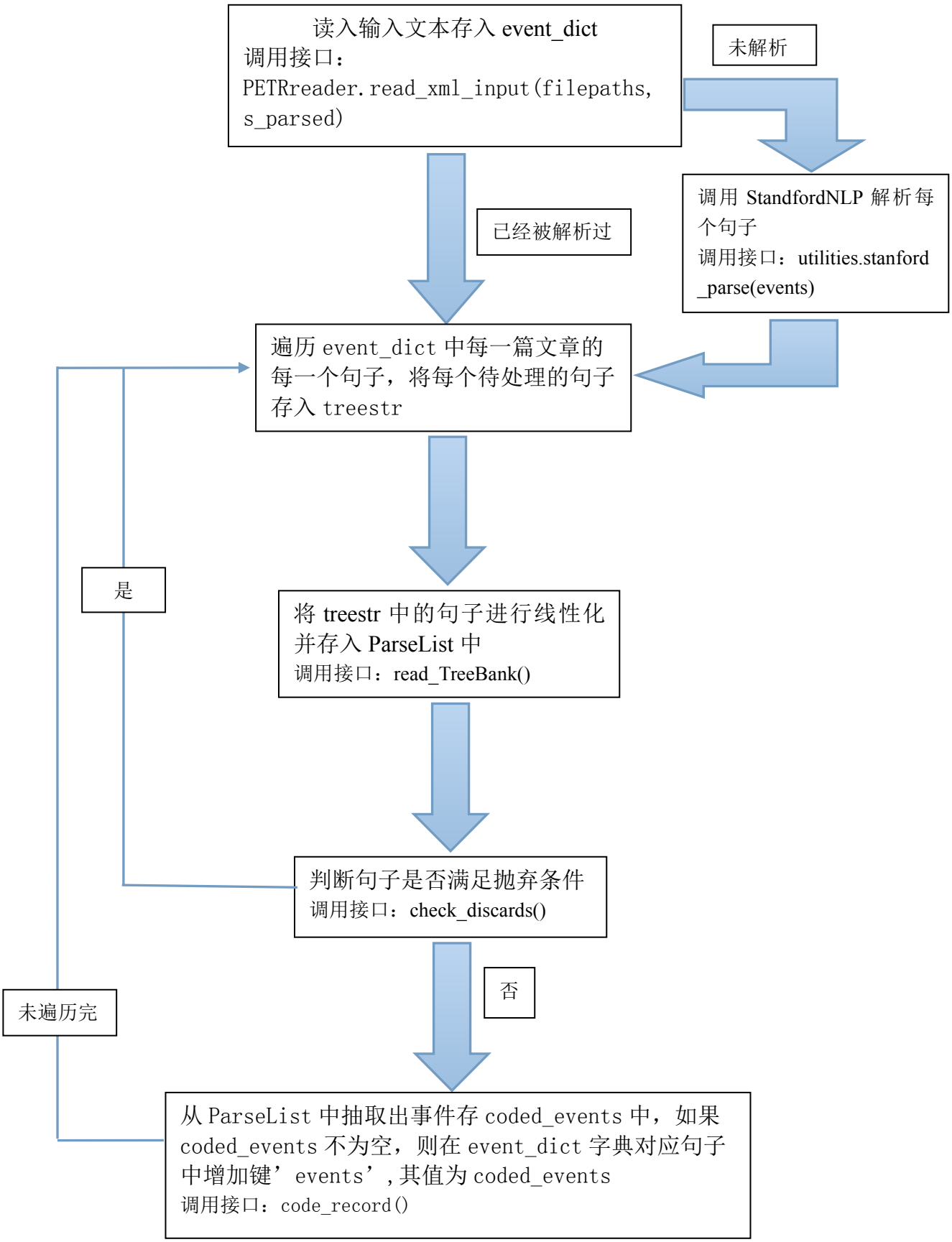
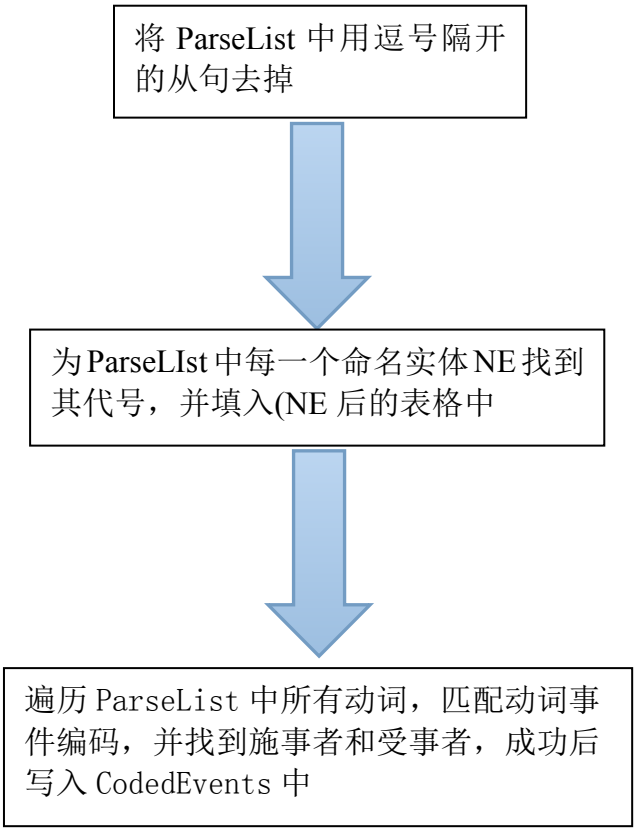
一、总流程图


读入输入文本存入 event_dict
调用接口：
PETRreader.read_xml_input(filepaths, s_parsed)
未解析
调用 StandfordNLP 解析每个句子
调用接口：utilities.stanford_parse(events)
已经被解析过
遍历 event_dict 中每一篇文章的每一个句子，将每个待处理的句子存入 treestr
将 treestr 中的句子进行线性化并存入 ParseList 中
调用接口：read_TreeBank()
判断句子是否满足抛弃条件
调用接口：check_discards()
是
否
未遍历完
从 ParseList 中抽取出事件存 coded_events 中，如果 coded_events 不为空，则在 event_dict 字典对应句子中增加键’events’,其值为 coded_events
调用接口：code_record()

二、code_record() 流程图



三、接口

1.read_verb_dictionary(verb_path)

功能：读入动词词典路径，将动词词典存储到字典 PETRglobals.VerbDict 中去

动词词典：同义词词典+动词同义词+动词匹配模型

同义词词典：&AUXVERB1_
+WAS
+WERE
+BEEN

动词同义词：--- KILL [190] ---
KILL
+SHOOT_DOWN {+SHOOTS_DOWN +SHOOTING_DOWN
+SHOT_DOWN }
+STAMP_OUT
+WIPE_OUT
MURDER [190]
SLAY {SLAYS SLAIN SLAYING}
STRANGLE [180]
BEHEAD
DECAPITATE
HANG {HANGS HANGING HANGED HUNG }
+STRING_UP {+STRUNG_UP }
+KIDNAPPED_AND_KILLED {} [180]
+KILLED_AND {}
+KILLED_OR {}

这里的+号表示后面跟的是一个动词词组，{}内是该动词的变位

动词模型（以-开头）：--- ATTEMPT [---] ---
ATTEMPT
- CITE ASSASSINATION * AS &EVIDENCE AGAINST + [112]
- + THREAT ASSASSINATION_* BY $ [138]
- + * HAVE BEEN JOKE $ SAID [111]
- * SETTLE &DISPUTE [040]
- * BREAK DEADLOCK [040]
- * VERIFY Y CLAIM [090]
- * ASSASSINATION [185]
- * KILL LEADER [185]
- * HALT BLOOD [040]
- * CONVINCE [020]
- BLOCK * [120]

{..}括号内内容表示动词变位，如 HANGS,HANGING,HANGED,HUNG 是动词 HANG 的变位；*代表动词所在位置；$代表施事者位置，如果$在动

词前面，那么程序将会只在动词前寻找施事者；+代表受事者位置；^符号代表忽略这个位置识别到的名词；&表示使用同义词词典，如上文中的 EVIDENCE 可以被它的同义词代换；_表示匹配动词时，两个单词必须连接在一起；%表示施事者有多个

调用接口后，动词同义词和动词匹配模型将被存储到字典当中。

-假如一个单词如 APPORTIONING 只是 ALLOCATE [060]的动词同义词，那么将被存储为

APPORTIONING -> ALLOCATE [060]

-假如一个单词如 FOUND 既是 DISCOVER 的动词同义词，又和 OUT 组成动词词组成为 HEAR 的动词同义词，那么 DISCOVER 被存储为

FOUND ::

[u'---', [u'040', u'HEAR ', (True, u'OUT')], [False, u'---', u'DISCOVER ']]

第一个 u'---'表示 FOUND 的编码，---表示无编码

[u'040', u'HEAR ', (True, u'OUT')]中的 TRUE 表明是一个词组，FOUND OUT 作为一个词组指向编码为 040 的 HEAR

[False, u'---', u'DISCOVER ']中的 FALSE 表示不是一个词组，FOUND 指向没有编码的 DISCOVER

-假如一个单词如 FASTEN 有其自己的动词模式匹配，则将被存储为

FASTEN ::

[u'---', [[u' ', u'SHIPS', u' ', u'&FIGHT'], [u' ', u'SOLIDARITY'], u'0312']]

这里的[u' ', u'SHIPS', u' ', u'&FIGHT']代表 FASTEN 之前的部分，[u' ', u'SOLIDARITY']表示 FASTEN 之后的部分

例子：词典中的

--- LOCK [---] --- （这里的 LOCK 是 FASTEN 的解释，没什么用）

FASTEN

INTERLOCK

+LOCK_IN

+LOCK_UP

ANCHOR

- &FIGHT SHIPS * SOLIDARITY [0312]

将会被做如下存储：

FASTEN ::

[u'---', [[u' ', u'SHIPS', u' ', u'&FIGHT'], [u' ', u'SOLIDARITY'], u'0312']]

INTERLOCKS -> FASTEN [---]

LOCK ::

[u'---', [u'---', u'FASTEN ', (True, u'UP ')], [u'---', u'FASTEN ', (True, u'IN ')]]

2.read_actor_dictionary(actorfile)

功能：读入主语词典路径，将主语词典存储到字典 PETRglobals.ActorDict 中去，并建立一个索引字典 PETRglobals.ActorCodes

主语词典：BELARUS_
+GOMEL_
+PINSK_
+LUBAVITCH_
[USRBLR <910825]
[BLRUNR 910825-911221]
[BLR >911221]

在这里，BELARUS,GOMEL,PINSK,LUBAVITCH 都代表同一个名词，处理时，当事件发生在 91 年 8 月 25 日前时，使用代号 USRBLR，同理当事件发生在 91 年 8 月 25 日至 91 年 12 月 21 日时使用代号 BLRUNR，发生在91年12月21日之后则使用代号BLR。时间的格式可以是YYYYMMDD或 YYMMDD，当 YY 小于 30 时默认为 20YY 年，否则则默认为 19YY 年。当然这个有关时间的选项是可以不要的，当时间对命名没有影响时可以写成：

BELARUS_ [BLR]
+GOMEL_
+PINSK_
+LUBAVITCH_

调用接口后，词典将会被存储到字典当中，指向同一个意思的一类词将会被赋予一个索引，如果这个词语有多个意思那么就将有多个索引，如：

AT&T ::
[[22244, u'_', (u'', u' ')]]

在这里，AT&T 的索引为 22244，这个索引在字典 PETRglobals.ActorCodes 中被存储为：

22244: ([u'MNCUSAMED'],)

通过查看索引，可以得到 AT&T 的代号为 MNCUSAMED

有多个索引的情况：

ITYOPPYA ::
[[5291, u'_', (u'', u' ')], [9711, u'_', (u'', u' ')]]

复合名词的情况：

SOMCHAI ::
[[16125, u'_', (u'WONGSAWAT', u'_'), (u'', u' ')]]

这里，复合名词为 SOMCHAI WONGSAWAT，指向索引 16125，通过查看索引

16125: ([2, 148714, 149019, u'THAGOV'],)

可知如果时间发生在 148714 和 149019 之间，那么主语代号为 THAGOV 否则主语代号为空，这里的时间使用的是 ANSI DATE，即 1601 年 1 月 1 日的的时间为 1，后面的依次类推。

3.read_discard_list(discard_path)

功能：读入筛选词路径，将筛选词读入 PETRglobals.DiscardList 列表中
筛选词：即起筛选功能的词，例如：

2011
2012
2013
MONTHS LATER
YEARS LATER
WEEKS LATER
+ World Golf Championships
+Manchester United
2014 COMPETITION

这个表的意思是，凡是含有 2011,2012，MONTHS LATER，2014 COMPETITION 等的句子都不要了。特别地，当词语前面有+号时表示只要含有这个词，那么整篇文章都不要了。

调用接口后，这个表将会以列表的形式存在 PETRglobals.DiscardList 中，每当处理一个段落之前便会使用这个列表检查有没有需要删除的句子。

4.read_dictionaries()

功能：调用以上的 read_verb_dictionary(verb_path)，read_actor_dictionary(actor file),read_discard_list(discard_path)接口,读入..\data\dictionaries 下的词典，储存成相应的结构。

5.run(filepaths, out_file, s_parsed)

功能：接受输入文件路径或路径列表 filepaths，输出文件路径 out_file，是否已经 parser 过 s_parsed 这三个参数，返回抽取出的事件列表，写入 out_file。

6.do_coding(event_dict, out_file)

功能：接受 parser 过的事件词典，抽取出事件，存入 event_dict 中
event_dict 输入格式（字典）：
(:::表示在字典中级别最高，::次之，以下以此类推)
AFP0808020824:::
sents::
1:
content
A Tunisian court has jailed a man for two years for helping young militants join

an armed Islamic group in Lebanon, his lawyer said Wednesday.
parsed
(ROOT (S (S (NP (DT A) (NNP TUNISIAN) (NN COURT)) (VP (VBZ HAS) (VP (VBN JAILED) (NP (NP (DT A) (NN MAN)) (PP (IN FOR) (NP (CD TWO) (NNS YEARS)))) (PP (IN FOR) (S (VP (VBG HELPING) (S (NP (JJ YOUNG) (NNS MILITANTS)) (VP (VB JOIN) (NP (NP (DT AN) (JJ ARMED) (JJ ISLAMIC) (NN GROUP)) (PP (IN IN) (NP (NNP LEBANON)))))))))))) (, ,) (NP (PRP$ HIS) (NN LAWYER)) (VP (VBD SAID) (NP (NNP WEDNESDAY))) (. .)))
meta::
date:
20080806
source:
AFP

Event_dict 输出格式（字典）：
AFP0808020824:::
sents::
1:
content
A Tunisian court has jailed a man for two years for helping young militants join an armed Islamic group in Lebanon, his lawyer said Wednesday.
parsed
(ROOT (S (S (NP (DT A) (NNP TUNISIAN) (NN COURT)) (VP (VBZ HAS) (VP (VBN JAILED) (NP (NP (DT A) (NN MAN)) (PP (IN FOR) (NP (CD TWO) (NNS YEARS)))) (PP (IN FOR) (S (VP (VBG HELPING) (S (NP (JJ YOUNG) (NNS MILITANTS)) (VP (VB JOIN) (NP (NP (DT AN) (JJ ARMED) (JJ ISLAMIC) (NN GROUP)) (PP (IN IN) (NP (NNP LEBANON)))))))))))) (, ,) (NP (PRP$ HIS) (NN LAWYER)) (VP (VBD SAID) (NP (NNP WEDNESDAY))) (. .)))
Issues
NAMED_THEROR_GROUP 1
Events
DZAREBMUS TUN 173
meta::
date:
20080806
source:
AFP

可以看到，相比于输入，输出多了’issues’与’events’这两个键。在这里，’issues’是可选项，可以人为制作一个 issues 的列表，用来从文章当中匹配列表中的词语。’events’是从这个句子中抽取出的事件列表，在本例中，只有一个事件被抽取了出来，施事者为 DZAREBMUS，受事者为 TUN，时间编码为 173。如句子中没有抽

取出’issues’或没有抽取出’events’则字典中没有相应键值。

7.read_TreeBank()

功能：将 treestr 里面存储的当前句子的树状语法结构改为线性结构保存在 ParseList 列表中，处理连接词，将语法树末尾的）改为~+对应的名字，如对应(NE 的）变为~NE，即

- 将名词短语变为命名实体并线性化，如：

  (NP (NP (DT A) (NN COURT)) (PP (IN IN) (NP (NNP GUYANA))))

  ---->

  (NE --- A COURT IN GUYANA ~NE

- 为动词短语添加索引，如：

  (VP (VBD SAID)

  ---->

  (VP1 (VBD SAID ~VBD

  表示 said 是句子里面第一个动词

- 去掉句子中的(POS’S),如：

  (NP (NP (NNP RWANDA) (POS ’S)) (NN INFORMATION) (NN MINISTER))

  ---->

  (NE --- RWANDA INFORMATIONMINISTER ~NE

- 处理连接词

  连接两个 NP 的在删除连接词，并前面加上 NEC 以及一个索引，如：

  (NP (NNP FRANCE) (CC AND) (NNP CHINA))

  ---->

(NEC1 (NE --- FRANCE ~NE (NE --- CHINA ~NE ~NEC1

  连接两个动词或句子，将(CC 改为(CCP，如：

  (CC BUT) #原句为 France and China on Wednesday accused Rwanda of making`` unacceptable accusations’’ by alleging Paris played an active role in the 1994 genocide, but said it was still determined to mend damaged ties with Kigali.

  ---->

  (CCP BUT ~CCP

这样，就完成了句子的线性化，下面是一个例子：

原语法树：

(ROOT (S (NP (DT THE) (NN STOPOVER)) (VP (VBD CAME) (SBAR (IN AS) (S (NP (DT THE) (NNP US) (NN LEADER)) (VP (VBD PREPARED) (S (VP (TO TO) (VP (VB ATTEND) (NP (NP (DT THE) (NNP BEIJING) (NNPS OLYMPICS)) (, ,)

```
(NP (NP (DT AN ) (NN EVENT ) ) (SBAR (WHNP (WDT WHICH ) ) (S (VP (MD WILL )
(VP (VB TEST ) (NP (PRP$ HIS ) (NN VOW ) (S (VP (TO TO ) (VP (VB KEEP ) (NP
(NNS POLITICS ) ) (PP (IN OUT ) (PP (IN OF ) (NP (DT THE ) (NNPS
GAMES ) ) ) ) ) ) ) ) ) ) ) ) ) ) ) ) ) ) ) ) )
(. . ) ) )
```

转化后列表（从左到右从上到下）：

(S	(NE	---	THE	STOPOVER	~NE	(VP1
(VBD	CAME	~VBD	(SBAR	(IN	AS	~IN
(S	(NE	---	THE	US	LEADER	~NE
(VP2	(VBD	PREPARED	~VBD	(S	(VP3	(TO
TO	~TO	(VP4	(VB	ATTEND	~VB	(NP1
(NE	---	THE	BEIJING	OLYMPICS	~NE	(,
,	~,	(NP2	(NE	---	AN	EVENT
~NE	(SBR	WHICH	WILL	TEST	HIS	VOW
TO	KEEP	POLITICS	OUT	OF	THE	GAMES
~SBR	~NP2	~NP1	~VP4	~VP3	~S	~VP2
~S	~SBAR	~VP1	(.	.	~.	~S
~ROOT						

8.code_record()

功能：从一个句子的线性化的语法树 ParseList 中抽取出事件，返回列表如:
[[u’ USA’ ,u’ CHN’ ,u’ 141’],[u’ GBROPP’ ,u’ CHN’ ,u’ 141’]]

9.assign_NEcodes()

功能：在前面将语法树线性化的时得到的 ParseList 中，(NE 后面跟了---，如：
[u’(NE’, u’---’, u’THE’, u’FRENCH’, u’MILITARY’, u’IN’, u’RWANDA’, u’~NE’]
调用这个接口，可以找到 NE 这个命名实体里的名词所对应的代号，并用其替代---，得：
[u’(NE’, u’FRAMIL’, u’THE’, u’FRENCH’, u’MILITARY’, u’IN’, u’RWANDA’, u’~NE’]
若没有找到对应代号，则保留---

10.check_verbs()

功能：从经过 assign_NEcodes()处理过的 ParseList 中提取出时间，存储在 CodedEvents 中。