# 基于专家系统的词性标注器

**13300200019 吴耀波**

## 1 问题描述

词性标注（Part-of-Speech tagging 或POS tagging），又称词类标注或者简称标注，是指为分词结果中的每个单词标注一个正确的词性的程序，也即确定每个词是名词、动词、形容词或其他词性的过程。

在该项目中，我们将要使用专家系统解决一个简单的词性标注问题，这里对输入的文本作一定约束：

- 输入文本仅含单个句子
- 句子为简单陈述句
- 不含标点符号

故我们可以把问题描述如下：

问题：对于简单陈述句进行词性标注

输入：不含标点符号的单句

输出：对于句中的每个单词，输出对应的词性

约束：使用专家系统解决

## 2 专家系统

### 2.1 事实库

初始事实库由三部分组成：词元、前后关系、词性标注。

- 词元即句中每一个单词，均作为一个事实加入到事实库，如“$i, he, say$”；
- 前后关系即对每一对相邻的单词记录其各自的词元以及前后关系作为一个事实，如“$i\ precedes\ have$”，对于首尾两个单词，添加“$BEGIN\_SENTENCE\ precedes\ \$word$”和“$\$word\ precedes\ END\_SENTENCE$”；
- 词性标注即把已标注的单词将其词元和词性作为一个事实加入到事实库，如“$good = Adjective$”。

### 2.2规则

在该系统中设计了两种规则：词表规则、关系规则。

- 词表规则为预设的词典，对于特定的单词制定相应的词性集，包括动词、副词、代名词、名词、形容词、介词、不定代词，如：
  - $went = Verb$
  - $quickly = Adverb$
  - $he = Pronoun$
  - $time = Noun$
  - $new = Adjective$
  - $to = Preposition$
  - $the = Determiner$
- 关系规则为根据某个词的词性和前后关系，推出另一个词的词性，如：
  - $BEGIN\_SENTENCE\ precedes\ \$word \rightarrow \$word = Noun\ or\ \$word = Pronoun$

- $\$word1\ precedes\ \$word2,\ \$word2 = Noun \rightarrow \$word1 = Adjective$
- $\$word1\ precedes\ \$word2,\ \$word2 = Verb \rightarrow \$word1 = Adverb$
- $\$word1\ precedes\ \$word2,\ \$word2\ precedes\ \$word3,\ \$word1 = Preposition,\ \$word2 = Determ$

**2.3演绎推理**

该系统使用前向链接推理，就是根据已有事实推断出新的事实。例如已知事实$A\ is\ x$，根据规则$A\ is\ x \rightarrow B\ is\ y$。获得$B\ is\ y$。然后将$B\ is\ y$加入数据库。再寻找新的规则，即$B\ is\ y \rightarrow ...$。

# 3 代码说明

在shell中运行Example.py即可运行示例程序：

```
python Example.py
```

或在ipython中加载RuleBaseExpertTagger.py，然后调用tagSentence(sentence)函数标注自定义句子：

```
#run in ipython
%run -i RuleBaseExpertTagger.py
tagSentence("i am a boy")
```

整个系统包含四个文件：

- **RuleBaseExpertTagger.py** #包含规则库、事实库、用户接口
  - **class** RuleBasedSystem #专家系统，封装规则库与事实库
  - **class** Rule #描述一个规则的类，完整的规则库如下：

```
verbIs = Rule('verb-is', ['is'], ['is = Verb'])
verbWent = Rule('verb-went', ['went'], ['went = Verb'])
verbRan = Rule('verb-ran', ['ran'], ['ran = Verb'])
verbBought = Rule('verb-bought', ['bought'], ['bought = Verb'])
verbGo = Rule('verb-go', ['go'], ['go = Verb'])
verbHave = Rule('verb-have', ['have'], ['have = Verb'])
verbBe = Rule('verb-be', ['be'], ['be = Verb'])
verbMake = Rule('verb-make', ['make'], ['make = Verb'])
verbSee = Rule('verb-see', ['see'], ['see = Verb'])

adverbQuickly = Rule('adverb-quickly', ['quickly'], ['quickly = Adverb'])
adverbLoudly = Rule('adverb-loudly', ['loudly'], ['loudly = Adverb'])
adverbEasily = Rule('adverb-Easily', ['easily'], ['easily = Adverb'])

pronounHe = Rule('pronoun-he', ['he'], ['he = Pronoun'])
pronounShe = Rule('pronoun-she', ['she'], ['she = Pronoun'])
pronounIt = Rule('pronoun-it', ['it'], ['it = Pronoun'])
pronounI = Rule('pronoun-i', ['i'], ['i = Pronoun'])
pronounThey = Rule('pronoun-they', ['they'], ['they = Pronoun'])
pronounWe = Rule('pronoun-we', ['we'], ['we = Pronoun'])
pronounThem = Rule('pronoun-them', ['them'], ['them = Pronoun'])
pronounThat = Rule('pronoun-that', ['that'], ['that = Pronoun'])

nounTime = Rule('noun-time', ['time'], ['time = Noun'])
nounFood = Rule('noun-food', ['food'], ['food = Noun'])
nounItem = Rule('noun-item', ['item'], ['item = Noun'])
```

```
nounPlace = Rule('noun-place', ['place'], ['place = Noun'])
nounStore = Rule('noun-store', ['store'], ['store = Noun'])
nounEssence = Rule('noun-essence', ['essence'], ['essence = Noun'])

adjectiveGood = Rule('adjective-good', ['good'], ['good = Adjective'])
adjectiveNew = Rule('adjective-new', ['new'], ['new = Adjective'])
adjectiveFirst = Rule('adjective-first', ['first'], ['first = Adjective'])
adjectiveBig = Rule('adjective-big', ['big'], ['big = Adjective'])
adjectiveOld = Rule('adjective-old', ['old'], ['old = Adjective'])
adjectiveLast = Rule('adjective-last', ['last'], ['last = Adjective'])
adjectiveBrobdingnagian = Rule('adjective-brobdingnagian', ['brobdingnagian'],
['brobdingnagian = Adjective'])

prepositionTo = Rule('preposition-to', ['to'], ['to = Preposition'])
prepositionOf = Rule('preposition-of', ['of'], ['of = Preposition'])
prepositionIn = Rule('preposition-in', ['in'], ['in = Preposition'])
prepositionFor = Rule('preposition-for', ['for'], ['for = Preposition'])
prepositionWith = Rule('preposition-with', ['with'], ['with = Preposition'])
prepositionOn = Rule('preposition-on', ['on'], ['on = Preposition'])
prepositionUp = Rule('preposition-up', ['up'], ['up = Preposition'])

determinerThe = Rule('determiner-the', ['the'], ['the = Determiner'])
determinerA = Rule('determiner-a', ['a'], ['a = Determiner'])
determinerAn = Rule('determiner-an', ['an'], ['an = Determiner'])
determinerNo = Rule('determiner-no', ['no'], ['no = Determiner'])
determinerThat = Rule('determiner-that', ['that'], ['that = Determiner'])

beginningSentence = Rule('begin-sentence', ['BEGIN_SENTENCE precedes ?word'],
['?word = Noun', '?word = Pronoun'])
adverbRule = Rule('adverb-placement', ['?word1 precedes ?word2', '?word2 =
Verb'], ['?word1 = Adverb'])
adjectiveRule1 = Rule('adjective-placement1', ['?word1 precedes ?word2', '?
word2 = Noun'], ['?word1 = Adjective'])
adjectiveRule2 = Rule('adjective-placement2', ['?word1 precedes ?word2', '?
word2 = Pronoun'], ['?word1 = Adjective'])
adjectiveRule3 = Rule('adjective-placement3', ['?word1 precedes ?word2', '?
word2 = Adjective'], ['?word1 = Adjective'])
adjectiveRule4 = Rule('adjective-placement4', ['?word1 precedes ?word2', '?
word1 = Verb'], ['?word2 = Adjective'])
determinerRule1 = Rule('determiner-placement1', ['?word1 precedes ?word2', '?
word2 = Noun'], ['?word1 = Determiner'])
determinerRule2 = Rule('determiner-placement2', ['?word1 precedes ?word2', '?
word2 = Pronoun'], ['?word1 = Determiner'])
nounRule1 = Rule('noun-placement1', ['?word1 precedes ?word2', '?word1 =
Determiner'], ['?word2 = Noun'])
nounRule2 = Rule('noun-placement2', ['?word1 precedes ?word2', '?word2 =
Verb'], ['?word1 = Noun'])
nounRule3 = Rule('noun-placement3', ['?word1 precedes ?word2', '?word2 precedes
?word3', '?word1 = Preposition', '?word2 = Determiner'], \
                    ['?word3 = Noun'])
pronounRule1 = Rule('pronoun-placement1', ['?word1 precedes ?word2', '?word2 =
Verb'], ['?word1 = Pronoun'])
pronounRule2 = Rule('pronoun-placement2', ['?word1 precedes ?word2', '?word2
precedes ?word3', '?word1 = Preposition', '?word2 = Determiner'], \
                    ['?word3 = Pronoun'])
```

```
verbRule1 = Rule('verb-placement1', ['?word1 precedes ?word2', '?word1 =
Noun'], ['?word2 = Verb'])
verbRule2 = Rule('verb-placement2', ['?word1 precedes ?word2', '?word1 =
Pronoun'], ['?word2 = Verb'])
verbRule3 = Rule('verb-placement3', ['?word1 precedes ?word2', '?word2 =
Preposition'], ['?word1 = Verb'])
```

    - **def** tagSentence(sentence) #用户接口，标记一个句子的词性

- **Engine.py** #推理引擎
    - **def** inferNewFacts(rules, workingMemory) #根据已有事实和规则不断产生新的事实，直至无新事实时返回
    - **def** matchAllRules(rules, workingMemory) #根据已有事实匹配所有规则，若匹配成功返回新事实
    - **def** mathRule(rule, workMemroy) #根据已用事实匹配所选规则，若匹配成功返回新事实
- **Utils.py** #字符串预处理工具库
- **Examples.py** #测试demo

# 4 结果展示

在Examples.py中测试该词性标注器的表现，总共10个句子，5个句子由已定义的词组成，另外5个有已定义和未定义的词混合组成。

```
# these sentences use only words that were defined
sentence1 = "time is of the essence"
sentence2 = "she quickly ran to the store"
sentence3 = "that is a big old place"
sentence4 = "i see that person"
sentence5 = "it is good that i went to the store"

# these sentences use a mix of words, some are defined and some are not
sentence6 = "i accidentally bought old food from the store"
sentence7 = "she went for another item"
sentence8 = "i have to see the walrus"
sentence9 = "he knows the store is too brobdingnagian"
sentence10 = "you have quickly moved up the ladder"
```

以下为输出的词性标注结果：

```
['time = Noun', 'is = Verb', 'of = Preposition', 'the = Determiner', 'essence =
Noun']
['she = Pronoun', 'quickly = Adverb', 'ran = Verb', 'to = Preposition', 'the =
Determiner', 'store = Noun']
['that = Pronoun', 'is = Verb', 'a = Determiner', 'big = Adjective', 'old =
Adjective', 'place = Noun']
['i = Pronoun', 'see = Verb', 'that = Pronoun', 'person = Noun']
['it = Pronoun', 'is = Verb', 'good = Adjective', 'that = Pronoun', 'i = Pronoun',
'went = Verb', 'to = Preposition', 'the = Determiner', 'store = Noun']

['i = Pronoun', 'accidentally = Adverb', 'bought = Verb', 'old = Adjective', 'food
= Noun', 'from = Verb', 'the = Determiner', 'store = Noun']
['she = Pronoun', 'went = Verb', 'for = Preposition', 'another = Adjective', 'item
= Noun']
['i = Pronoun', 'have = Verb', 'to = Preposition', 'see = Verb', 'the =
Determiner', 'walrus = Noun']
['he = Pronoun', 'knows = Verb', 'the = Determiner', 'store = Noun', 'is = Verb',
'too = Adjective', 'brobdingnagian = Adjective']
['you = Noun', 'have = Verb', 'quickly = Adverb', 'moved = Verb', 'up =
Preposition', 'the = Determiner', 'ladder = Noun']
```