第3章 词性标注

CHAPTER 3

词性标注(part-of-speech tagging,POS tagging)也被称为语法标注(grammatical tagging)或词类消疑(word-category disambiguation),是语料库语言学(corpus linguistics)中将语料库内单词的词性按其含义和上下文内容进行标记的文本数据处理技术。

词性标注可以由人工或特定算法完成,使用机器学习(machine learning)方法实现词性标注是自然语言处理(natural language processing,NLP)的研究内容。常见的词性标注算法包括隐马尔可夫模型(hidden Markov model,HMM)、条件随机场(conditional random field,CRF)等。词性标注主要被应用于文本挖掘(text mining)和 NLP 领域,是各类基于文本的机器学习任务(例如语义分析(semantic analysis)和指代消解(coreference resolution)的预处理步骤)。下面分别从原理和实战工具两个方面详细讲解。

3.1 词性标注原理

所谓词性标注,就是根据句子的上下文信息给句中的每个词确定一个最为合适的词性标记。比如,给定一个句子:“我中了一张彩票”。对该句子的标注结果可以是:“我/代词/中/动词/了/助词/一/数词/张/量词/彩票/名词。/标点”。

词性标注的难点主要是由词性兼类所引起的。词性兼类是指自然语言中某个词语的词性多于一个的语言现象,它是自然语言中的普遍现象,例如句子:S1="他是山西大学的教授。"和 S2="他在山西大学教授计算语言学。"在句子 S1 中,“教授”是一个表示职称的名词,而在句子 S2 中“教授”是一个动词。对人而言,这样的词性歧义现象比较容易排除,但是对于没有先验知识的机器而言是比较困难的。词性兼类在中文中很突出,据不完全统计,常见的词性兼类现象有几十种,这些兼类现象具有以下分布特征:

(1) 在汉语词汇中,兼类词的数量不多,占总词条的 5%～11%。

(2) 兼类词的实际使用频率很高,占总词条的 40%～45%。也就是说,越是常用的词,其词性兼类现象越严重。

(3) 兼类词现象分布不均:在孙茂松等的统计中,仅动名兼类就占全部兼类现象的 49.8%;在张民门的统计中,动名兼类和形副兼类占全部 113 种兼类现象的 62.5%。词性兼类的消歧常采用概率的方法,如隐马尔可夫模型。这些方法的有效性依赖于兼类词性的概率分布。但是有些兼类词性的概率分布近似,特别是高频的词性兼类现象,如中文的动词名词兼类,对于这些兼类现象,传统的概率方法很难奏效,如何解决这个问题是目前词性标注面临的主要困难之一。

3.1.1 词性介绍[8]

词性指以词的特点作为划分词类的根据。词类是一个语言学术语，是一种语言中词的语法分类，是以语法特征（包括句法功能和形态变化）为主要依据、兼顾词汇意义对词进行划分的结果，现代汉语的词可以分为 13 种词类。从组合和聚合关系来说，词类是指：在一个语言中，众多具有相同句法功能、能在同样的组合位置中出现的词，聚合在一起形成的范畴。词类是最普遍的语法的聚合。词类划分具有层次性。如中文中，词可以分成实词和虚词，实词中又包括体词、谓词等，体词中又可以分出名词和代词等。

1. 词类区分

词类根据表示的实际意义以及语法结构可以分为实词和虚词，按照是否吸收其他词性的词可分为开放词类和闭合词类（例如，中文的动词可以直接作为“某种动作的名字”当成名词使用，所以中文的名词是一个开放词类）。以上的大类还可以按照词的具体用法和功能分为小类。

2. 实词

实词是表示具体概念的词，实词可以分为：

1）名词

名词是指表示实体和概念名称的词。在大多数屈折语中，名词具有以下性质。

- 性：对于大多数印欧语言都分（或部分地分）阴，阳，中；一些小语种用“动物性”或“非动物性”区分词性，如格鲁吉亚语。某些语言还有更多分类方式，或交叉地采用上述分类方式。
- 数：表示物体是单个、特定的几个或多个，即单数或复数。有些语种包括双数等特定数的词法。
- 格：表示名词在句子中的成分，即主格（第一格）、与格（第二格）、属格（第三格）、宾格（第四格）等。部分语言如希腊语、俄语等还有更多的词格。

在屈折语中，需要注意主谓一致，即谓语的形式需要根据主语的性和数屈折变化。

2）代词

代词是指在句子结构中代替其他词的词，包括人称代词（代替某一人称或事物的词，如“你”“我”“他”）、疑问代词（包括“5W1H”）、指示代词（“这”“那”等）。代表名词的代词通常也具有名词的性、数、格规律。

3）动词

动词是表示动作的词。根据是否带宾语可以分为“及物动词”与“不及物动词”，“及物动词”还包括“双宾语动词（“他给了我一块糖中”的“给”需要“我”和“糖”两个宾语）”和“双及物动词（需要宾语和补语的动词，例如“他觉得我很好”需要“我”这个宾语和“很好”这个描述性的补语）”。有些语言存在不需要主语的动词（尤其是表示天气的词如“下雪了”这一说法中，英语必须有 it 这个主语，中文和西班牙语则不需要），有些涉及“交易”的动词需要 3 个宾语。

表示“某种动作的名称”的词称作“动名词”，在某些语言中有特定的词法。

在屈折语中，动词根据时态（过去时、现在时、将来时、一般动作、进行时、完成时及其交叉）和语态（主动、被动）变化。

4）形容词

形容词是用来修饰名词，表示人或事物的性质、状态、特征或属性的词。在屈折语中形容词根据所修饰的词语性质屈折变化。

5）数词

数词是表示数量（基数词）和序数（序数词）的词。

6）量词

量词（measure word/numeral classifier/counter word）是表示数量单位的词。中文和日语在大多数描述数量的语境下都使用“数词＋量词”构成的数量短语。

量词下面还分为“数量词”（表示可数名词数量单位的词，如“个”“条”等）、“体量词”（表示一个整体的不可数名词的数量单位的词，如“堆”）和“动量词”（表示动作次数的词，如“下”“次”等）。

英文对不特定数目的物体使用“集合名词”，如“一叠纸”（a stack of paper）中的“叠”属于集合名词。

7）区别词

区别词是一类不能单独充当谓语的“形容词”，即不能不加助词地组成“S 是 V”句子的形容词。每个区别词通常有一个反义词，表示互相对立的两种属性之一。区别词通常可以后加“的”组成“的字短语”作为谓语。

3. 虚词

虚词泛指没有完整意义，但有语法意义或功能的词。具有必须依附于实词或语句、表示语法意义，不能单独成句，不能单独作语法成分，不能重叠的特点。虚词有以下几种：

1）副词

副词是修饰动词，表示动作的特征、状态等的词。有些副词是形容词变化而来的，实际地表示动作的特征状态等（如中文中大多数“形容词＋地”格式的副词短语和英文中以“形容词＋ly”构成的副词），有些副词特别地构成句法成分。

2）介词

介词是用在句子的名词成分之前，说明该成分与句子其他成分关系的词。

3）连词

连词是连接两句话，表示其中逻辑关系的词。

4）助词

助词是表示语气、句子结构和时态等语法和逻辑性的“小词”。在有词语曲折的语言中助词一般不曲折。

5）叹词

叹词是表示感叹的小词，通常独立成句。不少粗话都以叹词的形式独立存在。

6）拟声词

拟声词是模拟声音的小词，如“砰”“啪”等。英语中某些拟声词同时也是表示这种声音的名词，如“roar”既是模仿动物吼声的拟声词，又是名词“吼叫”。

对词性了解后，下一步就需要了解怎样从一个完整的句子中把词性标注和识别出来，这就会用到算法，下面介绍 3 种算法：HMM、感知器、CRF。

3.1.2　HMM 词性标注[9]

HMM 也就是隐马尔可夫模型，我们再回顾一下其原理，HMM 是结构最简单的动态贝叶斯网络。描述由一个隐藏的马尔可夫链随机生成不可观测的状态随机序列，再由各个状态生成一个观测而产生随机序列的过程。隐藏的马尔可夫链随机生成的状态的序列称为状态序列，每个状态生成一个观测，称为观测序列。

HMM 做了两个基本的假设：

齐次马尔可夫假设，即假设隐藏的马尔可夫链在任意时刻 t 的状态只依赖于前一时刻的状态，与其他观测状态无关。

观测独立性假设，即假设任意时刻的观测只依赖于该时刻的马尔可夫链的状态，与其他观测以及状态无关。

HMM 由初始状态概率向量$\boldsymbol{\pi}$ 、状态转移概率矩阵 $\boldsymbol{A}$ 以及观测概率矩阵 $\boldsymbol{B}$ 决定。

在词性标注问题中，初始状态概率为每个语句序列开头出现的词性的概率，状态转移概率矩阵由相邻两个单词的词性得到，观测序列为分词后的单词序列，状态序列为每个单词的词性，观测概率矩阵 $\boldsymbol{B}$ 也就是一个词性到单词的概率矩阵。

HMM 有 3 个基本问题：

概率计算问题，给出模型和观测序列，计算在模型 λ 下观测序列 O 出现的概率。

学习问题，估计模型 $\lambda=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$ 的参数，使得该模型下观测序列概率 $P(O|\lambda)$ 最大，也就是用极大似然的方法估计参数。

观测问题，已知模型 λ 和观测序列 O，求给定的观测序列条件概率 $P(I|O)$ 下最大的状态序列 I，即给定观测序列，求最可能的状态序列。

在词性标注问题中，需要解决的是学习问题和观测问题。学习问题即转移矩阵的构建，观测问题即根据单词序列得到对应的词性标注序列。

接下来用 Python 代码实现一个例子。

小明和小芳是两个城市的学生，现在小明知道小芳在下雨天时待在家看电视的概率为 60%，出去逛街的概率为 10%，洗衣服的概率为 30%；晴天时洗衣服的概率为 45%，出去逛街的概率为 50%，在家看电视的概率为 5%；阴天时出去逛街的概率为 55%，洗衣服的概率为 30%，看电视的概率为 15%。那么 3 天内小芳出现“逛街-洗衣服-看电视”的情况的概率是多少？

HMM 的构成主要有以下 5 个参数：

(1) 模型的状态数(晴天，雨天，阴天)。

(2) 模型的观测数(逛街，洗衣，看电视)。

(3) 模型的状态转移概率矩阵 $\boldsymbol{A}_{ij}$，表示 step $n-1$ 到 step n，状态 i 下一个是状态 j 的概率。

(4) 模型的状态发射概率矩阵 $\boldsymbol{B}_i(k)$ 表示 i 状态下产生 k 观测状态的概率。

(5) $\boldsymbol{\pi}$ 为初始状态分布概率矩阵。

首先求解上面的参数，代码如下所示。

```
def ChinesePOS():
    #转移概率 Aij,t 时刻由状态 i 变为状态 j 的频率
    #观测概率 Bj(k),由状态 j 观测为 k 的概率
    #PAI i 初始状态 q 出现的频率
    #先验概率矩阵
    pi = {}
    a = {}
    b = {}
#所有的词语
    ww = {}
#所有的词性
    pos = []
    #每个词性出现的频率
    frep = {}
```

```
#每个词出现的频率
frew = {}
fin = codecs.open("处理语料.txt","r","utf8")
for line in fin.readlines():
    temp = line.strip().split(" ")
    for i in range(1,len(temp)):
        word = temp[i].split("/")
        if len(word) == 2:
            if word[0] not in ww:
                ww[word[0]] = 1

            if word[1] not in pos:
                pos.append(word[1])
fin.close()
ww = ww.keys()
for i in pos:
    #初始化相关参数
    pi[i] = 0
    frep[i] = 0
    a[i] = {}
    b[i] = {}
    for j in pos:
        a[i][j] = 0
    for j in ww:
        b[i][j] = 0
for w in ww:
    frew[w] = 0
line_num = 0
#计算概率矩阵
fin = codecs.open("处理语料.txt","r","utf8")
for line in fin.readlines():
    if line == "\n":
        continue
    tmp = line.strip().split(" ")
    n = len(tmp)
    line_num += 1
    for i in range(1,n):
        word = tmp[i].split("/")
        pre = tmp[i-1].split("/")
        #计算词性频率和词频率
        frew[word[0]] += 1
        frep[word[1]] += 1
        if i == 1:
            pi[word[1]] += 1
        else :
            a[pre[1]][word[1]] += 1
        b[word[1]][word[0]] += 1
for i in pos:
    #计算各个词性的初始概率
    pi[i] = float(pi[i])/line_num
    for j in pos:
        if a[i][j] == 0:
            a[i][j] = 0.5
    for j in ww:
        if b[i][j] == 0:
            b[i][j] = 0.5
for i in pos:
    for j in pos:
        #求状态 i 的转移概率分布
```

```
            a[i][j] = float(a[i][j])/(frep[i])
        for j in ww:
            #求词 j 的发射概率分布
            b[i][j] = float(b[i][j])/(frew[j])
    return a,b,pi,pos,frew,frep
    print "game over"
```

参数求解完毕，运用动态规划的 Viterbi 算法求解最佳路径，代码如下所示：

```
def viterbi(a,b,pi,str_token,pos,frew,frep):
    # dp = {}
    #计算文本长度
    num = len(str_token)
    #绘制概率转移路径
    dp = [{} for i in range(0,num)]
    #状态转移路径
    pre = [{} for i in range(0,num)]
    for k in pos:
        for j in range(num):
            dp[j][k] = 0
            pre[j][k] = ''
#句子初始化状态概率分布(首个词在所有词性的概率分布)
    for p in pos:

        if b[p].has_key(str_token[0]):

            dp[0][p] = pi[p] * b[p][str_token[0]] * 1000
        else:
            dp[0][p] = pi[p] * 0.5 * 1000
    for i in range(0,num):
        for j in pos:
            if (b[j].has_key(str_token[i])):
                sep = b[j][str_token[i]] * 1000
            else:
                #计算发射概率,这个词不存在,应该置 0.5/frew[str_token[i]],这里默认为 1
                sep = 0.5 * 1000
            for k in pos:
                #计算本 step i 的状态是 j 的最佳概率和 step i-1 的最佳状态 k(计算结果为 step i
                #所有可能状态的最佳概率与其对应 step i-1 的最优状态)
                if (dp[i][j]< dp[i-1][k] * a[k][j] * sep):

                    dp[i][j] = dp[i-1][k] * a[k][j] * sep
                    #各个 step 最优状态转移路径
                    pre[i][j] = k
    resp = {}
    #
    max_state = ""
    #首先找到最后输出的最大观测值的状态设置为 max_state
    for j in pos:
        if max_state == "" or dp[num-1][j] > dp[num-1][max_state]:
            max_state = j
    # print
    i = num -1
#根据最大观测值 max_state 和前面求的 pre 找到概率最大的一条
    while i>=0:
        resp[i] = max_state
        max_state = pre[i][max_state]
        i -= 1
    for i in range(0,num):
        print str_token[i] + "\\" + resp[i].encode("utf8")
```

```
#主入口函数代码:
if __name__ == "__main__":
    a,b,pi,pos,frew,frep = ChinesePOS()
    #北京/ns 举行/v 新年/t 音乐会/n
    str_token = [u"北京",u"举行",u"新年",u"音乐会"]
    viterbi(a, b, pi, str_token, pos, frew,frep)
```

运行结果如下：

```
北京\ns
举行\v
新年\t
音乐会\n
```

3.1.3　感知器词性标注[10]

按照中文分词时的经验，感知器能够利用丰富的上下文特征，是优于 HMM 的选择，对于词性标注也是如此。

感知器词性标注示例如代码 3.1 所示。

【代码 3.1】　train_perceptron_pos.py

```
from pyhanlp import *
import zipfile
import os
from pyhanlp.static import download, remove_file, HANLP_DATA_PATH
def test_data_path():
    """
    获取测试数据路径,位于 $ root/data/test,根目录由配置文件指定。
    :return:
    """
    data_path = os.path.join(HANLP_DATA_PATH, 'test')
    if not os.path.isdir(data_path):
        os.mkdir(data_path)
    return data_path
## 验证是否存在 MSR 语料库,如果没有则自动下载
def ensure_data(data_name, data_url):
    root_path = test_data_path()
    dest_path = os.path.join(root_path, data_name)
    if os.path.exists(dest_path):
        return dest_path

    if data_url.endswith('.zip'):
        dest_path += '.zip'
    download(data_url, dest_path)
    if data_url.endswith('.zip'):
        with zipfile.ZipFile(dest_path, "r") as archive:
            archive.extractall(root_path)
        remove_file(dest_path)
        dest_path = dest_path[:-len('.zip')]
    return dest_path
## 指定 PKU 语料库
PKU98 = ensure_data("pku98", "http://file.hankcs.com/corpus/pku98.zip")
PKU199801 = os.path.join(PKU98, '199801.txt')
PKU199801_TRAIN = os.path.join(PKU98, '199801-train.txt')
PKU199801_TEST = os.path.join(PKU98, '199801-test.txt')
POS_MODEL = os.path.join(PKU98, 'pos.bin')
NER_MODEL = os.path.join(PKU98, 'ner.bin')
## ================================================
```

```
## 以下开始 感知机 词性标注
AbstractLexicalAnalyzer = JClass('com.hankcs.hanlp.tokenizer.lexical.AbstractLexicalAnalyzer')
PerceptronSegmenter = JClass('com.hankcs.hanlp.model.perceptron.PerceptronSegmenter')
POSTrainer = JClass('com.hankcs.hanlp.model.perceptron.POSTrainer')
PerceptronPOSTagger = JClass('com.hankcs.hanlp.model.perceptron.PerceptronPOSTagger')

def train_perceptron_pos(corpus):
    trainer = POSTrainer()
    trainer.train(corpus, POS_MODEL)                          # 训练感知机模型
    tagger = PerceptronPOSTagger(POS_MODEL)                   # 加载
    analyzer = AbstractLexicalAnalyzer(PerceptronSegmenter(), tagger)  # 构造词法分析器,
    # 与感知机分词器结合,能同时进行分词和词性标注。
    print(analyzer.analyze("李狗蛋的希望是希望上学"))                  # 分词 + 词性标注
    print(analyzer.analyze("李狗蛋的希望是希望上学").translateLabels())  # 对词性进行翻译
    return tagger

if __name__ == '__main__':
    train_perceptron_pos(PKU199801_TRAIN)
```

运行速度会有些慢,结果如下:

```
李狗蛋/nr 的/u 希望/n 是/v 希望/v 上学/v
李狗蛋/人名 的/助词 希望/名词 是/动词 希望/动词 上学/动词
```

这次的运行结果完全正确,感知器成功地识别出未登录词“李狗蛋”的词性。

3.1.4 CRF 词性标注[11]

介绍中文分词时讲到过 CRF,CRF 词性标注示例如代码 3.2 所示。

【代码 3.2】 train_crf_pos.py

```
from  pyhanlp import *
import zipfile
import os
from pyhanlp.static import download, remove_file, HANLP_DATA_PATH

def test_data_path():
    """
    获取测试数据路径,位于 $root/data/test,根目录由配置文件指定。
    :return:
    """
    data_path = os.path.join(HANLP_DATA_PATH, 'test')
    if not os.path.isdir(data_path):
        os.mkdir(data_path)
    return data_path
## 验证是否存在 MSR 语料库,如果没有则自动下载
def ensure_data(data_name, data_url):
    root_path = test_data_path()
    dest_path = os.path.join(root_path, data_name)
    if os.path.exists(dest_path):
        return dest_path
    if data_url.endswith('.zip'):
        dest_path += '.zip'
    download(data_url, dest_path)
    if data_url.endswith('.zip'):
        with zipfile.ZipFile(dest_path, "r") as archive:
            archive.extractall(root_path)
        remove_file(dest_path)
        dest_path = dest_path[:-len('.zip')]
    return dest_path
```

```
## 指定 PKU 语料库
PKU98 = ensure_data("pku98", "http://file.hankcs.com/corpus/pku98.zip")
PKU199801 = os.path.join(PKU98, '199801.txt')
PKU199801_TRAIN = os.path.join(PKU98, '199801 - train.txt')
PKU199801_TEST = os.path.join(PKU98, '199801 - test.txt')
POS_MODEL = os.path.join(PKU98, 'pos.bin')
NER_MODEL = os.path.join(PKU98, 'ner.bin')
## ================================================
## 以下开始 CRF 词性标注
AbstractLexicalAnalyzer = JClass('com.hankcs.hanlp.tokenizer.lexical.AbstractLexicalAnalyzer')
PerceptronSegmenter = JClass('com.hankcs.hanlp.model.perceptron.PerceptronSegmenter')
CRFPOSTagger = JClass('com.hankcs.hanlp.model.crf.CRFPOSTagger')
def train_crf_pos(corpus):
    # 选项 1.使用 HanLP 的 Java API 训练,慢
    tagger = CRFPOSTagger(None)                                    # 创建空白标注器
    tagger.train(corpus, POS_MODEL)                                # 训练
    tagger = CRFPOSTagger(POS_MODEL)                               # 加载
    # 选项 2.使用 CRF++训练,HanLP 加载。(训练命令由选项 1 给出)
    # tagger = CRFPOSTagger(POS_MODEL + ".txt")
    analyzer = AbstractLexicalAnalyzer(PerceptronSegmenter(), tagger)  # 构造词法分析器,
    # 与感知机分词器结合,能同时进行分词和词性标注。
    print(analyzer.analyze("李狗蛋的希望是希望上学"))       # 分词 + 词性标注
    print(analyzer.analyze("李狗蛋的希望是希望上学").translateLabels())  # 对词性进行翻译
    return tagger
if __name__ == '__main__':
    tagger = train_crf_pos(PKU199801_TRAIN)
```

运行时间会比较长,结果如下:

```
李狗蛋/nr 的/u 希望/n 是/v 希望/v 上学/v
李狗蛋/人名 的/助词 希望/名词 是/动词 希望/动词 上学/动词
```

CRF 词性标注依然可以成功识别未登录词“李狗蛋”的词性。

了解词性标注的原理和算法后,下面介绍流行的开源工具 Jieba 和 HanLP,用它们来做词性标注。

3.2 词性标注工具实战

在中文分词的章节中已经介绍过 Jieba 和 HanLP 开源工具,Python 语言中最流行的分词工具就是 Jieba,它不仅能做分词,在得到分词结果的同时还会返回对应的词性。HanLP 也一样,在分词的同时默认返回词性。

3.2.1 Python 的 Jieba 词性标注

Jieba 可以返回分词结果以及对应的词性,示例如代码 3.3 所示。

【代码 3.3】 jieba_pos.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
#__author__ = '陈敬雷'
import jieba
import jieba.analyse
import jieba.posseg

def dosegment_all(sentence):
    '''
```

```
    带词性标注,对句子进行分词,不排除停用词等
    :param sentence:输入字符
    :return:
    ''
'
    sentence_seged = jieba.posseg.cut(sentence.strip())
    outstr = ''
    for x in sentence_seged:
        outstr += "{}/{},".format(x.word,x.flag)
    return outstr

str = dosegment_all("充电了么App是专注上班族职业技能提升充电学习的在线教育平台")
print("词性标注输出结果如下:")
print(str)
'''
词性标注输出结果如下:
充电/v,了/ul,么/y,App/eng,是/v,专注/v,上班族/nz,职业技能/n,提升/v,充电/v,学习/v,的/uj,在线教育/l,平台/n,
'
'
'
```

Jieba 词性标注如表 3.1 所示。

表 3.1 Jieba 词性标注

英 文	中 文	举 例	数 量
a	形容词	高 明 尖 诚 粗陋 冗杂 丰盛 顽皮 很贵 挺好用 ……	4306
ad	副形词	努目 完全 努力 切面 严实 慌忙 明确 仓皇 详细 ……	110
ag	形语素	详 笃 睦 奇 洋 裸 渺 忤 虐 黢 怠 峻 悫 鄙 秀 ……	46
an	名形词	麻生 猥琐 腐生 困苦 危难 负疚 刚愎 危险 悲苦 ……	40
b	区别词	劣等 洲际性 超常规 同一性 年级 非农业 二合一 ……	1363
c	连词	再者说 倘 只此 或曰 以外 换句话说 虽是 除非 ……	504
d	副词	幸免 四顾 绝对 急速 特约 从早 务虚 逐行 挨边 ……	2422
df	不要	不要	1
dg	副语素	俱 辄	2
e	叹词	好哟 嗄 天呀 哎 哇呀 啊哈 嗳 诶 嗬 呜呼 ……	34
f	方位词	内侧 以来 面部 后侧 面前 沿街 之内 两岸 里 ……	351
g	语素	媸 璇 戬 瓠 踔 鳖 撄 絮 腠 遘 醢 槊 胂 鹌 豳 ……	969
h	前接成分	非 超低	2
i	成语	绿荫蔽日 震耳欲聋 沧海一粟 一望无边 为尊者讳 ……	25583
j	简称略语	交警 中低收入 四个现代 经检测 青委 车改 ……	1396
k	后接成分	型 者 式 们	4
l	习用语	不懂装懂 相聚一刻 由下而上 十字路口 查无此人 ……	17721
m	数词	九六 十二 半成 戊酉 俩 一二三四五 丙戌 片片 ……	13178
mg	数语素	寅 巳	2
mq	数量词	半年度 四方面 十副 三色 一口钟 四面 三分钟 ……	80
n	名词	男性 娇子 气压 写实性 联立方程 商业智能 寒窗 ……	117902
ng	名语素	诀 卉 茗 鹊 娃 寨 酊 钬 雹 役 莺 谊 隙 族 鸩 ……	280
nr	人名	雍正皇帝 小老弟 唐僧骑 铁娘子 小甜甜 璐 ……	72842
nrfg	古近代人名	刘备 关羽 张飞 赵云 任弼时 ……	484
nrt	音译人名	米尔科 达尼丁 三世 五丁 塞拉 埃克尔斯 贝当 ……	5941
ns	地名	南明 锡山 拱北 南非 哥里 平北 丹井 佛山 广州 ……	17706

续表

英 文	中 文	举 例	数 量
nt	机构团体	浙江队 中医院 中华网 铁道部 广电部 联想集团 ……	4713
nz	其他专名	培根 补丁 圣战士 英属 国药准字 ……	10441
o	拟声词	哈喇 咝 咔嚓 飕 哇哇 喃 咕隆 咿呀 叽咕 ……	247
p	介词	顺当 顺着 借了 连着 乘着 除了 较之于 根 自 ……	114
q	量词	毫厘 盅 封 千瓦小时 立方米 盎 座 毫克 张 斛 ……	232
r	代词	该车 这时 那些 什么 鄙人 此案 睿智者 他 怎生 ……	759
rg	代语素	兹	1
rr	代词	你们 其他人	3
rz	代词	这位	1
s	处所词	世外 肩前 舷外 手下 耳边 兜里 盘头 桌边 家外 ……	591
t	时间词	新一代 清时 先上去 月初 昔年 无日 唐五代 佳日 ……	1768
tg	时间语素	昔 晚 春 现 暮 夕 宵	7
u	助词	则否 等 恁地 等等 似的 来说 矣哉 来看 般 的话 ……	20
ud	得	得	1
ug	过	过	1
uj	的	的	1
ul	了	了	1
uv	地	地	1
uz	着	着	1
v	动词	批发 孕育 做成 纳闷儿 遭殃 留话 吻下去 创生 ……	34761
vd	副动词	狡辩 持续 逆势	3
vg	动语素	悖 谏 踞 泯 濯 掳 谄 疑 诲 吁 囿 酌 蟠 豢 匿 ……	160
vi	动词	沉溺于 等同于 沉湎于 徜徉于	4
vn	名动词	审查 相互毗连 销蚀 对联 劳工 漫游 ……	3235
vq	动词	挨过 念过 去过 去净	4
x	非语素字	舭 珑 娄 躅 蕺 蜓 螂 窀 蘅 荬 姆 榍 飑 楂 ……	367
y	语气词	吓呆了 呃 呀 兮 哩 呐 嘞 哇 呗 意味着 也罢 啦 ……	49
z	状态词	歪曲 飘飘 慢慢儿 急地 沉迷在 晕乎乎 ……	2624
zg	zg	鲚 瑑 灉 鄌 缘 嗙 獘 洅 鬲 埄 涚 鞞 檬 肸 撎 ……	5666

3.2.2 Java 的 HanLP 词性标注

HanLP 默认的分词结果就是带有词性标注的，示例如代码 3.4 所示。

【代码 3.4】 HanLPPosDemo.java

```
package com.chongdianleme.job;
import com.hankcs.hanlp.HanLP;
import com.hankcs.hanlp.seg.common.Term;
import java.util.List;
/**
 * Created by "充电了么"App - 陈敬雷
 * "充电了么"App 官网: http://chongdianleme.com/
 * "充电了么"App - 专注上班族职业技能提升充电学习的在线教育平台
 * HanLP 中文分词功能演示,开源地址: https://github.com/hankcs/HanLP
 */
public class HanLPPosDemo {
    public static void main(String[] args) {
        segment();//常用默认分词: HanLP.segment
```

```
    }
    /**
     * 常用默认分词：HanLP.segment
     * HanLP对词典的数据结构进行了长期的优化，可以应对绝大多数场景。哪怕HanLP的词典上百兆也无须担心，因为在内存中被精心压缩过。
     * 如果内存非常有限，请使用小词典。HanLP默认使用大词典，同时提供小词典。全部词典和模型都是惰性加载的，不使用的模型相当于不存在，可以自由删除。
     * HanLP.segment其实是对StandardTokenizer.segment标准分词的包装，和标准分词的结果是一样的。
     */
    public static void segment() {
        String s = "分布式机器学习实战(人工智能科学与技术丛书)深入浅出，逐步讲解分布式机器学习的框架及应用配套个性化推荐算法系统、人脸识别、对话机器人等实战项目。";
        List<Term> termList = HanLP.segment(s);
        System.out.println(termList);
        //输出结果如下：分词结果包含词性，比如分布式的词性b代表区别词，机器学习的词性gi代
        //表计算机相关词汇，实战的词性n代表名称，后面每个词都返回了对应的词性，这里不一一
        //举例，下章单独介绍词性标注，列出所有的词性表。
        //[分布式/b, 机器学习/gi, 实战/n, (/w, 人工智能/n, 科学/n, 与/p, 技术/n, 丛书/n, )/w,
        //深入浅出/i, ,/w, 逐步/d, 讲解/v, 分布式/b, 机器学习/gi, 的/uj, 框架/n, 及/c, 应用/vn,
        //配套/a, 个性化/v, 推荐/v, 算法/n, 系统/n, 、/w, 人脸/n, 识别/v, 、/w, 对话/vn, 机器
        //人/n, 等/u, 实战/n, 项目/n, 。/w]
    }
}
```

HanLP的词性标注如表3.2所示。

表3.2 HanLP词性标注

字母	描　述	字母	描　述
a	形容词	nr2	蒙古姓名
f	方位词	begin	仅用于始##始
mq	数量词	gi	计算机相关词汇
nn	工作相关名词	nf	食品，比如“薯片”
ad	副形词	nrf	音译人名
g	学术词汇	bg	区别语素
n	名词	gm	数学相关词汇
nnd	职业	ng	名词性语素
ag	形容词性语素	nrj	日语人名
gb	生物相关词汇	bl	区别词性惯用语
nb	生物名	gp	物理相关词汇
nnt	职务职称	nh	医药、疾病等健康相关名词
al	形容词性惯用语	ns	地名
gbc	生物类别	c	连词
nba	动物名	h	前缀
nr	人名	nhd	疾病
an	名词	nsf	音译地名
gc	化学相关词汇	cc	并列连词
nbc	动物纲目	i	成语
nrl	复姓	nhm	药品
b	区别词	nt	机构团体名
gg	地理、地质相关词汇	d	副词
nbp	植物名	j	简称略语

续表

字母	描 述	字母	描 述
ni	机构相关(不是独立机构名)	uls	来讲 来说 而言 说来
ntc	公司名	wb	百分号、千分号,全角:% ‰ 半角:%
dg	辄、俱、复之类的副词	pbei	介词“被”
k	后缀	s	处所词
nic	下属机构	usuo	所
ntcb	银行	wd	逗号,全角:,半角:,
dl	连语	q	量词
l	习用语	t	时间词
nis	机构后缀	uv	连词
ntcf	工厂	wf	分号,全角:;半角:;
e	叹词	qg	量词语素
m	数词	tg	时间词性语素
nit	教育相关机构	uyy	一样 一般 似的 般
ntch	酒店宾馆	wh	单位符号,全角:￥ $ £ ° ℃ 半角:$
end	仅用于终##终	qt	时量词
mg	数语素	u	助词
nl	名词性惯用语	uz	着
nth	医院	wj	句号,全角:。
nts	中小学	qv	动量词
Mg	甲、乙、丙、丁之类的数词	ud	助词
nm	物品名	uzhe	着
nto	政府机构	wky	右括号,全角:)〕］｝》】 〗〉半角:)]{>
ntu	大学	r	代词
ryt	时间疑问代词	ude1	的 底
nmc	化学品名	uzhi	之
vn	名动词	wkz	左括号,全角:(〔［｛《【〖〈半角:([{<
nx	字母专名	rg	代词性语素
ryv	谓词性疑问代词	ude2	地
uj	助词	v	动词
vshi	动词“是”	wm	冒号,全角:：半角::
nz	其他专名	Rg	古汉语代词性语素
rz	指示代词	ude3	得
ul	连词	vd	副动词
vx	形式动词	wn	顿号,全角:、
o	拟声词	rr	人称代词
rzs	处所指示代词	udeng	等 等等 云云
ule	了 喽	vf	趋向动词
vyou	动词“有”	wp	破折号,全角:——－－———半角:— —-
p	介词		
rzt	时间指示代词	ry	疑问代词
ulian	连(如,连小学生都会)	udh	的话
w	标点符号	vg	动词性语素
pba	介词“把”	ws	省略号,全角:…… …
rzv	谓词性指示代词	rys	处所疑问代词

续表

字母	描　　述	字母	描　　述
ug	过	wyz	左引号，全角：“ ‘『
vi	不及物动词(内动词)	yg	语气语素
wt	叹号，全角：！	x	字符串
wyy	右引号，全角：”’』	z	状态词
y	语气词(delete yg)	xu	网址 URL
vl	动词性惯用语	zg	状态词
ww	问号，全角：？	xx	非语素字