自然语言处理基础

# 隐马尔可夫模型

## 三个基本问题

## 三要素和模型训练

## 预测问题和序列标注

# 最大熵模型

## 基本思想

## 约束条件

## 预测表达式

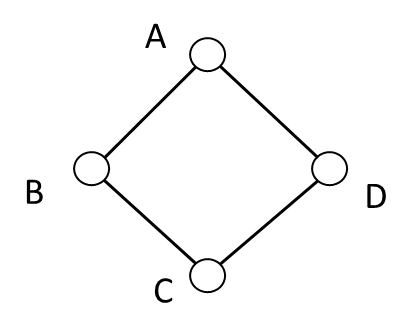
## 训练算法

# 条件随机场模型

## 数学原理

### 概率无向图模型——马尔可夫网络

（1）图的概率意义

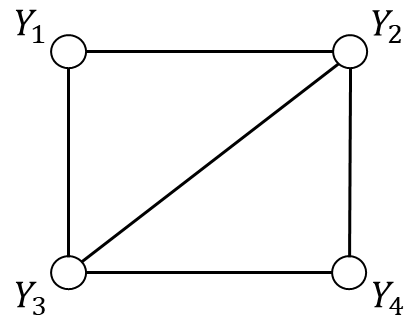


上图就是一个马尔可夫网络，图中的每个节点表示一个随机变量，相连节点表示两个随机变量相关，不相连的节点表示两个随机变量独立，于是可以写出下边的两个等式

（2）团：一个图中的任何两个节点之间都有边相连时，这个图就是一个团。

（3）因子函数（势函数或者能量函数）：定义在团c上的势函数。

（4）图的概率计算



一个图的所有随机变量的联合概率分布等于图中所有团上的势函数的乘积，即。

### HMM等生成模型存在的问题

（一）计算繁琐

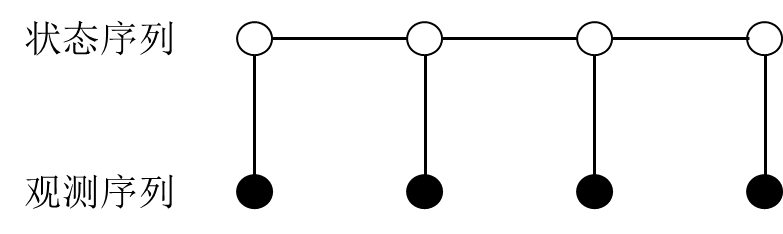
1. HMM是生成模型，需要先计算联合概率分布也就是发射概率矩阵。

2. 观测值是符号也就是离散值，因此不适用取值连续的观测。

3.必须计算每个状态发射所有可能观测符号的概率。

（二）观测符号是原子的

每个观测符号的概率与其前后的观测符号无关，如下图所示



（三）很难加入字词顺序以外的特征

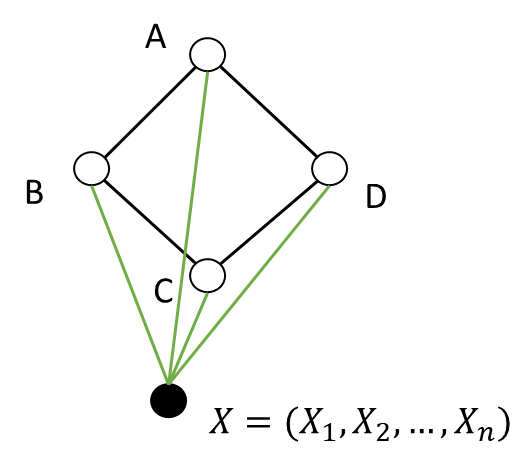
以实体识别为例，我们除了使用词语串作为特征以外，无法加入词性串特征。

（四）标注偏置问题

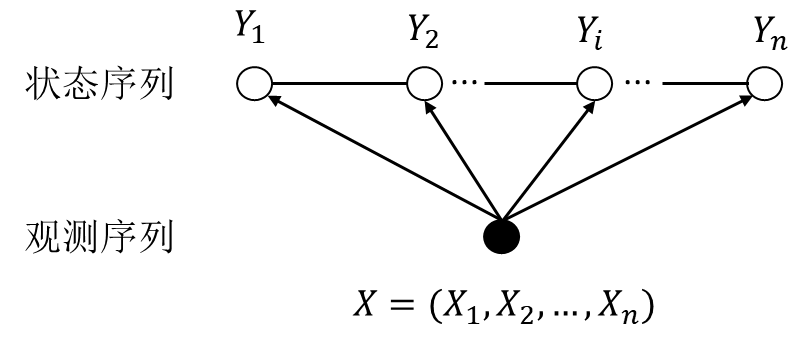
假设有两个相似的观测序列ri和ro，它们的状态序列分别为12和34，如果训练集中ri出现的次数远大于ro出现的次数，则预测观测序列ro对应的状态序列时，很容易将状态序列错误的预测为12。

### 条件随机场的定义

条件随机场就是在马尔可夫网络中加入输入节点，如下图所示



在自然语言处理中，使用的是线性链条件随机场，如下图所示



仿照着马尔可夫网络的联合概率分布公式可以写出条件随机场的条件概率分布公式，也就是模型预测公式

这里的就是输入的一个观测序列，就是对应的一个状态序列。表示序列中的位置序号，表示特征函数的序号，表示特征函数的序号。从3种序号来看的值取决于观测序列中每个位置（对求和）的两类特征函数（两类特征函数）各自拥有的所有特征函数（对求和，对求和）值。和就是模型中需要训练的未知数，表示对应特征函数的重要性，和越大表示特征在训练语料中出现频次越大，重要性越大。为归一化因子，它确保计算出的数值在0到1之间。和就是势函数也就是特征函数，它的取值为0或1。注意特征函数的自变量，说明该特征函数是定义在边（团）上的特征函数，是**局部**特征函数。注意特征函数的自变量，说明该特征函数是定义在结点上的特征函数，也是**局部**特征函数。从特征函数的所有自变量可以看出，该特征函数值取决于当前的序列位置，当前序列位置的状态，前一序列位置的状态，输入的序列。从特征函数的所有自变量可以看出，该特征函数值取决于当前的序列位置，当前序列位置的状态，输入的序列。

下面通过一个例子来理解上边的公式

例：设有一标注问题：输入观测序列为，输出标记序列为，取之于。假设特征，和对应的权值为，如下：

**对给定的观测序列，求标记序列为的非规范化条件概率。**

解：

为了计算列出下表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

上表中只有绿色高亮的元素才有值，其他元素都为0，因此

为了计算列出下表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

上表中只有绿色高亮的元素才有值，其他元素都为0，因此

### 简化形式的定义

其中为全局特征向量，一旦输入的观测序列和预测序列确定后，就成为有具体元素值的特征向量。是一个维的特征向量，即，的值等于特征函数的数量。是每个特征重要性的权重信息向量，显示。这里为什么说是全局特征向量呢，原因是它的每一个元素都是对整个序列来求值的，即，显然特征函数的定义最多只能涉及两个状态，当然最少也可以涉及一个状态。注意是整个观测序列。

## 特点

## 基本实践

### CRFsuite

### sklearn-CRFsuite

本节介绍基于sklearn-crfsuite实现一个简单的实体识别系统。

（1）准备数据

这里使用NLTK的CoNLL2002西班牙语语料。

|  |
| --- |
| # 查看训练和测试数据，这里数据使用NLTK中CoNLL2002的西班牙语料  print("文件ID", nltk.corpus.conll2002.fileids())  # 读出训练句子序列  train\_sents = list(nltk.corpus.conll2002.iob\_sents('esp.train'))  # 读出测试句子序列  test\_sents = list(nltk.corpus.conll2002.iob\_sents('esp.testb'))  # 1个句子由单词序列构成，每个的单词是一个三元组（词形，词性，实体标记）  print("打印训练集第1个句子", train\_sents[0]) |

（2）抽取特征

抽取特征的工作使用函数实现，这具有典型NLTK风格。通常条件随机场库（比如crf++）使用模板文件+语料生成特征。

|  |
| --- |
| # 这种定义一个函数来抽取特征的编码方式非常符合NLTK风格  # 给定一个句子的单词元组序列和当前词位置，返回针对该当前词的特征字典  def word2features(sent, i):  word = sent[i][0]  postag = sent[i][1]  features = {  'bias': 1.0,  'word.lower()': word.lower(),  'word[-3:]': word[-3:],  'word[-2:]': word[-2:],  'word.isupper()': word.isupper(),  'word.istitle()': word.istitle(),  'word.isdigit()': word.isdigit(),  'postag': postag,  'postag[:2]': postag[:2],  }  if i > 0:  word1 = sent[i-1][0]  postag1 = sent[i-1][1]  features.update({  '-1:word.lower()': word1.lower(),  '-1:word.istitle()': word1.istitle(),  '-1:word.isupper()': word1.isupper(),  '-1:postag': postag1,  '-1:postag[:2]': postag1[:2],  })  else:  features['BOS'] = True  if i < len(sent)-1:  word1 = sent[i+1][0]  postag1 = sent[i+1][1]  features.update({  '+1:word.lower()': word1.lower(),  '+1:word.istitle()': word1.istitle(),  '+1:word.isupper()': word1.isupper(),  '+1:postag': postag1,  '+1:postag[:2]': postag1[:2],  })  else:  features['EOS'] = True  return features  # 对一个句子中的每个单词抽取一个特征字典，返回所有单词的特征构成的特征序列  def sent2features(sent):  return [word2features(sent, i) for i in range(len(sent))]  # 返回一个句子（单词序列）的实体标记序列  def sent2labels(sent):  return [label for token, postag, label in sent] |

（3）训练

|  |
| --- |
| # 抽取训练数据每个单词的特征字典，构成特征字典序列X\_train  X\_train = [sent2features(s) for s in train\_sents]  # 训练数据每个特征字典对应的标记构成训练标记序列y\_train  y\_train = [sent2labels(s) for s in train\_sents]  # 抽取测试数据每个单词的特征字典，构成特征字典序列X\_test  X\_test = [sent2features(s) for s in test\_sents]  # 测试数据每个特征字典对应的标记构成测试标记序列y\_test  y\_test = [sent2labels(s) for s in test\_sents] |

（4）评测

|  |
| --- |
| y\_pred = crf.predict(X\_test)  labels = list(crf.classes\_)  # group B and I results  sorted\_labels = sorted(  labels,  key=lambda name: (name[1:], name[0])  )  # 从评测函数的输入参数可以看出：  # 实际标记序列y\_test与预测标记序列y\_pred必须在相同位置上必须对应的是同一个词语，  # 即分词一致  print(metrics.flat\_classification\_report(  y\_test, y\_pred, labels=sorted\_labels, digits=3  )) |

官网还介绍了如何输出重要的特征函数，这里就不再介绍了。

### CRF++

CRF++是一款简单的、可定制的、开源的CRF实现工具。它可用来进行分词和序列标注。CRF++是一款相对通用的工具，可用来处理各种NLP任务，如**命名实体识别、信息抽取、块提取**等。

#### 特征

（1）可以通过模板定义特征集

（2）使用C++的STL编写

（3）实现**LBFGS快速训练算法**，实现**拟牛顿法**解决数值型数据的训练问题。

（4）内存占用小

（5）输出top-n最优标注结果

（6）输出边缘概率分布

（7）Singe-best MIRA训练

（8）源码开源

#### 下载

2013年2月发布最新版代码。提供源码包和可执行程序包。

下载地址：<https://code.google.com/p/crfpp/downloads/list>。

#### 安装

（1）环境要求：gcc3.0及以上

（2）安装步骤：

|  |
| --- |
| $ ./configure  $ make  $ sudo make install |

#### 用法

1. 训练和测试文件的格式

（1）行的要求：每行一个token。多个token组成一个句子，句子之间以空行分隔。

（2）列的要求：每个token可以有多个列，但是各行列数必须一致，各列之间以空白分隔。每一列代表一种词法信息标记，比如第1列通常填字（词）形串，第2列填词性串等。

（3）标记列要求：标记列必须为最后一列。

示例：

|  |
| --- |
| He PRP B-NP  reckons VBZ B-VP  the DT B-NP  current JJ I-NP  account NN I-NP  deficit NN I-NP  will MD B-VP  narrow VB I-VP  to TO B-PP  only RB B-NP  # # I-NP  1.8 CD I-NP  billion CD I-NP  in IN B-PP  September NNP B-NP  . . O  He PRP B-NP  reckons VBZ B-VP  .. |

2. 编写特征模板

CRF++是一种通用序列标注工具，因此必须事先定义自己的模板，该模板用于描述训练和测试数据所要抽取的特征。

（1）基础模板和宏

模板文件中的每一行表示一种模板，一个模板的宏%x[row, col]用来指定输入数据中某个位置的一个token。row表示相对于当前token的行索引，col表示相对于当前token的列索引。

|  |
| --- |
| Input: Data  He PRP B-NP  reckons VBZ B-VP  the DT B-NP << CURRENT TOKEN  current JJ I-NP  account NN I-NP |

（2）模板类型

模板类型的标识在每行模板的行首，分别用字母“U”或字母“B”标识。其中字母“U”表示unigram模板，字母“B”表示bigram模板。需要注意的是unigram和bigram是针对输出类别标记的，而不是token。

下边举一个unigram的示例：

假设给定模板"U01:%x[0,1]"，和下边的一段tokens，

|  |
| --- |
| He PRP B-NP  reckons VBZ B-VP  the DT B-NP << CURRENT TOKEN  current JJ I-NP  account NN I-NP |

CRF++自动生成下边的特征函数集

|  |
| --- |
| func1 = if (output = B-NP and feature="U01:DT") return 1 else return 0  func2 = if (output = I-NP and feature="U01:DT") return 1 else return 0  func3 = if (output = O and feature="U01:DT") return 1 else return 0 |

显然，对于1个unigram模板，生成的特征函数数量为L\*N，其中L表示预测标记集合的长度，N表示对应于该模板展开的唯一token串的种类数。

bigram模板会将当前token和前一token组合起来展开，1个bigram模板生成的特征函数数量为L\*L\*N，其中L表示预测标记集合的长度，N表示对应于该模板展开的唯一token串的种类数。需要注意bigram模板会使训练和预测时间变长。

（3）模板标识

有时不同模板会展开成相同的token串，为了区分这一情况引入模板的编号。举例如下，输入tokens段为

|  |
| --- |
| The DT B-NP  pen NN I-NP  is VB B-VP << CURRENT TOKEN  a DT B-NP |

将"%x[-2,1]" 和"%x[1,1]"应用到该输入token段，都会展成"DT"这个值，但是它们来自不同的模板。为了区分两个值，在两个模板前加入模板表示号如下所示

|  |
| --- |
| U**01**:%x[-2,1]  U**02**:%x[1,1] |

（4）示例

下边列出一个完整的模板文件示例

|  |
| --- |
| # Unigram  U00:%x[-2,0]  U01:%x[-1,0]  U02:%x[0,0]  U03:%x[1,0]  U04:%x[2,0]  U05:%x[-1,0]/%x[0,0]  U06:%x[0,0]/%x[1,0]  U10:%x[-2,1]  U11:%x[-1,1]  U12:%x[0,1]q  U13:%x[1,1]  U14:%x[2,1]  U15:%x[-2,1]/%x[-1,1]  U16:%x[-1,1]/%x[0,1]  U17:%x[0,1]/%x[1,1]  U18:%x[1,1]/%x[2,1]  U20:%x[-2,1]/%x[-1,1]/%x[0,1]  U21:%x[-1,1]/%x[0,1]/%x[1,1]  U22:%x[0,1]/%x[1,1]/%x[2,1]  # Bigram  B |

3. 训练（编码问题）

（1）训练命令

|  |
| --- |
| $ **crf\_learn -t template\_file train\_file model\_file**  $ crf\_learn -f 3 -c 1.5 template\_file train\_file model\_file |

（2）命令参数

-a CRF-L2|CRF-L1：设置正则化方法，默认值为CRF-L2。一般情况下，L2效果优于L1，但是L1模型的非零特征远小于L2。

-c 浮点数：控制模型过拟合与欠拟合的程度，该值会对系统效果有显著的影响。该值越大，模型的过拟合程度越大。

-f 整数：特征选择的阈值，默认值为1。那些出现频次低于该阈值的特征将被删除。

-p 整数：对于多核CPU可以使用多线程完成模型训练，该参数指定线程数量。

（3）控制台输出信息

iter：当前迭代的次数

terr：token错误率

serr：sentence错误率

obj：当前模型对象值，当其收敛到一个固定值后，迭代停止

diff：本次迭代与前次迭代对象值的差

4. 预测（解码问题）

（1）预测命令

|  |
| --- |
| $ **crf\_test -m model\_file test\_files ... >result.txt** |

（2）命令参数

-v 1：预测结果中给出top1标记的概率

-v 2：预测结果中给出所有标记的概率

-n 整数：预测结果中给出topn标记的概率

#### 示例

在CRF++工具中的example目录中提供了4个示例，分别为基本名词块识别（basenp），组块分析（chunking），日本命名实体识别（JapaneseNE），分词（seg）。可以直接运行下边的命令进行训练和预测

|  |
| --- |
| $ crf\_learn template train model  $ crf\_test -m model test |

#### 提示

（1）CRF++的输入数据格式与YamCha工具

<http://chasen.org/~taku/software/yamcha/>

的输入数据格式相同，可以用同一份评测数据在两种工具上运行，对比CRF与SVM的识别效果。

（2）CRF++输出数据格式与CoNLL2002评测数据格式一致，可以用perl脚本

<https://www.aflat.org/conll2000/chunking/output.html>

评测系统效果，该脚本将给出所有标记类的F值。

#### 参考

（1）J. Lafferty, A. McCallum, and F. Pereira. **Conditional random fields: Probabilistic models for segmenting and labeling sequence data**, In Proc. of ICML, pp.282-289, 2001

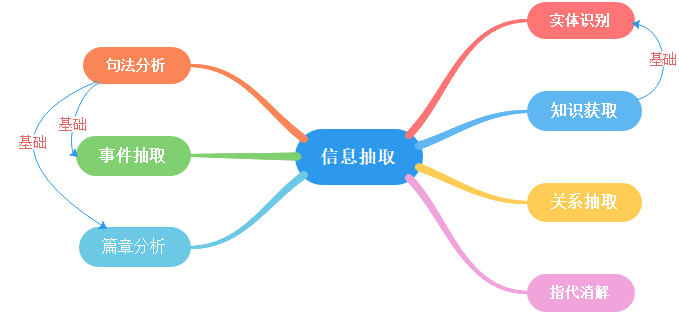
（2）F. Sha and F. Pereira. **Shallow parsing with conditional random fields**, In Proc. of HLT/NAACL 2003

# 信息抽取

## 概念

## 任务

郭喜跃在《信息抽取研究综述》中指出信息抽取的任务包括（1）实体识别（2）指代消解（3）关系抽取（4）事件抽取。李保利在《信息抽取研究综述》中指出信息抽取包括（1）实体识别（2）句法分析（3）篇章分析（4）知识获取。从以上两位的综述得到下边的信息抽取任务导图：



## 与相关概念之间的关系

## 研究现状

## 评测与语料

### CoNLL-2002

只是英文的命名实体识别语料

<https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus>

### CoNLL-2003

<https://www.clips.uantwerpen.be/conll2003/ner/>

### CCSK2017

中文电子病历中的实体识别语料，包括的实体有疾病、症状、检查等。

<https://pan.baidu.com/s/1kVADq9D>

### 中文微博

<https://github.com/hltcoe/golden-horse>

## 工具调研

### Standford NER

官网地址 <https://nlp.stanford.edu/software/CRF-NER.shtml>

github地址 <https://github.com/Lynten/stanford-corenlp>

支持中文的命名实体识别。

### sklearn-crfsuite

sklearn-crfsuite是基于CRFsuite库的一款轻量级的CRF库。该库兼容sklearn的算法，因此可以结合sklearn库的算法设计实体识别系统。sklearn-crfsuite不仅提供了条件随机场的训练和预测方法还提供了评测方法。这里再简单说明一下CRFsuite。CRFsuite使用C/C++实现了条件随机场模型。作者为东京大学的Naoaki Okazaki，从他的个人信息页来看，他应该从事文本处理研究十多年了。从Github源码来看，作者将具体的实现算法全部封装成了动态库，这样可能不利于我们了解条件随机场内部原理。在众多实现条件随机场的软件中，CRFsuite具有下边的特征：

* 快速的训练和标注
* 简单的数据格式
* 工业级训练方法，具体实现的方法有：
  + 受限存储空间条件下的BFGS（L-BFGS）
  + Orthant-Wise受限存储空间条件下的拟牛顿法
  + 随机梯度下降法
  + 平均感知机法
  + 消极对抗法
  + 权重向量自适应正则化方法
* 数值归一化的前向后向算法
* 线性链条件随机场（一阶马尔可夫）
* 评测方法，自动计算precision，recall，F1值

需要说明的是：该评测方法要求文本原标记序列和预测标记序列对应位置词语必须相同，也就是说评测是针对分词后的词语序列来说的。

* 高效模型存储格式，用来存储/访问模型
* C++/SWIG API，提供C++语言接口，以及在C++语言之上的其他语言接口

CRFsuite官网地址为<http://www.chokkan.org/software/crfsuite/>

CRFsuite的Github地址为：<https://github.com/chokkan/crfsuite>

sklearn-crfsuite基础教程地址：

<https://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.html#evaluation>

# 汉语分词

## 概述

## 基于词典的分词方法

## 基于语料库的分词方法

## HanLP分词

### 分词器配置变量

分词器的相关配置定义在Config.java类中，这里我们将分词相关的所有配置变量列于下表

|  |  |  |  |
| --- | --- | --- | --- |
| 变量名称 | 变量名 | 默认值 | 备注 |
| 索引分词的词语最小长度 | indexMode | 0 | 整数值，为0时表示不进行索引全切分 |
| 是否识别中国人名 | nameRecognize | true |  |
| 是否识别音译人名 | translatedNameRecognize | true |  |
| 是否识别日本人名 | japaneseNameRecognize | false |  |
| 是否识别地名 | placeRecognize | false |  |
| 是否识别机构 | organizationRecognize | false |  |
| 是否加载用户词典 | useCustomDictionary | false |  |
| 用户词典是否为高优先级 | forceCustomDictionary | false |  |
| 是否进行词性标注 | speechTagging | false |  |
| 命名实体识别是否至少有一项被激活 | ner | true |  |
| 是否计算词语偏移量 | offset | false |  |
| 是否识别数字和量词 | numberQuantifierRecognize | false |  |
| 并行分词的线程数 | threadNumber | 1 |  |

这种配置类什么时候实例化呢，不用想肯定是分词开始前就会实例化，拿HanLP类中的ViterbiSegment分词类举例。该类的继承关系用如下图所示：



由继承关系图可以看到，只要实例化ViterbiSegment则首先会执行Segment()初始化，在该方法中实例化分词器配置对象config。这些配置变量都是公有变量，因此可以在ViterbiSegment类实例化为对象后直接在外部修改。那么什么时候来使用这些配置变量呢，当然是在分词的时候，具体是哪个类的哪个方法呢，当然是ViterbiSegment类的List<Term> segSentence(char[] sentence)方法。

### 词典的使用条件和先后顺序

我们知道了词典配置变量使用的位置后，就可以确定每个词典的使用条件了以及每个词典的使用顺序

1. 词语粗分

（1）构建词图

对应方法为void generateWordNet(final WordNet wordNetStorage)，在此方法中系统使用CoreNatureDictionary.txt文件切分出所有可能的分词路径。此时如果配置变量useCustomDictionary为true，则将CustomDictionary.txt中的词也考虑进来，说明CustomDictionary.txt优先级会高。另外大家可以看到CoreNatureDictionary.txt实际上也充当了隐马词性标注的发射矩阵，里边某些多词性词也列出了词性序列以及各词性对应的频次。

（2）用户定制词典干预

如果配置变量useCustomDictionary为true，即需要使用CustomDictionary.txt进行干预，则执行下边对应的方法，否则跳过该步骤。用户词典干预根据是否进行全切分有两种不同方法：当配置变量indexMode>0时，即系统处于全切分模式时，对应方法为

List<Vertex> combineByCustomDictionary(List<Vertex> vertexList, DoubleArrayTrie<CoreDictionary.Attribute> dat, final WordNet wordNetAll)，

如果indexMode=0，即系统处于普通分词模式，对应方法为

List<Vertex> combineByCustomDictionary(List<Vertex> vertexList, DoubleArrayTrie<CoreDictionary.Attribute> dat)。

从调用的方法我们不难看出，全切分时系统会根据CustomDictionary.txt添加分词路径。而普通切分时，系统会根据CustomDictionary.txt合并路径。这也就是为什么有的时候明明已经在CustomDictionary.txt中添加了新词却不生效的原因，因为一旦根据CoreNatureDictionary.txt构建了词图就不会再有新的路径插到已有分词路径中间，此时就去查找并修改CoreNatureDictionary.txt中的相关字或词吧。

（3）维特比选择最优路径

对应方法为List<Vertex> viterbi(WordNet wordNet)，至此就得到了一个粗分的分词结果。需要注意HanLP的Viterbi分词只是用viterbi方法求解最优路径，并不是隐马。

3. 数字识别

如果配置变量numberQuantifierRecognize为true，则在粗分结果的基础上进行数字合并操作，否则直接跳过该步。对应方法为

void mergeNumberQuantifier(List<Vertex> termList, WordNet wordNetAll, Config config)。

4. 实体识别

配置变量ner为true时，则需要进行各种实体的识别，继续向下执行。需要注意该变量受其他实体识别变量影响，只要其他任意实体配置变量为true，则ner就会为true。如果ner为false，则跳过下边各项实体识别继续词性标注环节。

（1）中国人名识别

执行此步，配置变量nameRecognize必须为true。调用方法为

PersonRecognition.recognition(vertexList, wordNetOptimum, wordNetAll)。人名使用隐马，因此有转移矩阵nr.tr.txt和发射矩阵nr.txt。由于HanLP不提供训练语料，我们自己也很难得到有角色标注的语料，因此我们一般只修改nr.txt文件，删除nr.txt.bin文件后生效。

（2）音译人名识别

执行此步，配置变量translatedNameRecognize必须为true。调用方法为

TranslatedPersonRecognition.recognition(vertexList, wordNetOptimum, wordNetAll)。需要注意音译人名的识别没有用隐马，就是匹配分词法。涉及到的词典为nrf.txt，如果用户修改该词典，则需要删除nrf.txt.trie.dat使其生效。

（3）日本人名识别

执行此步，配置变量japaneseNameRecognize必须为true。调用方法为

JapanesePersonRecognition.recognition(vertexList, wordNetOptimum, wordNetAll)。需要注意日本人名的识别没有用隐马，就是匹配分词法。涉及到的词典为nrj.txt，如果用户修改该词典，则需要删除nrj.txt.trie.dat和nrj.txt.value.dat使其生效。

（4）地名识别

执行此步，配置变量placeRecognize必须为true。调用方法为

PlaceRecognition.recognition(vertexList, wordNetOptimum, wordNetAll)。地名使用隐马，因此有转移矩阵ns.tr.txt和发射矩阵ns.txt。由于HanLP不提供训练语料，我们自己也很难得到有角色标注的语料，因此我们一般只修改ns.txt文件，删除ns.txt.bin文件后生效。

（5）机构名识别

执行此步，配置变量organizationRecognize必须为true。调用方法为

OrganizationRecognition.recognition(vertexList, wordNetOptimum, wordNetAll)。注意这里在调用机构名识别之前先进行了一次识别，也就是层叠隐马，而人名和地名的识别就是普通的隐马。机构名的识别使用层叠隐马，涉及的文件有转移矩阵nt.tr.txt和发射矩阵nt.txt。由于HanLP不提供训练语料，我们自己也很难得到有角色标注的语料，因此我们一般只修改nt.txt文件，删除ns.txt.bin文件后生效。机构名的识别需要人名地名识别具有较高准确率。

|  |
| --- |
| 其他没有在系统中使用的词典有  机构名词典.txt  全国地名大全.txt  人名词典.txt  上海地名.txt  现代汉语补充词库.txt  这些词典是对系统中的词典的更新记录，如果你添加了新的人名、地名、机构名可以在这里添加保存。  另外，如果需要添加人名、地名、机构名可以直接在CoreNatureDictionary.txt中添加，最好是3字以上实体，如果要去掉错误识别的命名实体可以直接在相应的nr.txt，ns.txt，nt.txt中添加。 |

5. 索引全切分

如果配置变量indexMode为true则使用方法

List<Term> decorateResultForIndexMode(List<Vertex> vertexList, WordNet wordNetAll)执行全切分。

6. 词性标注

如果配置变量speechTagging为true则进行隐马词性标注，对应方法为

void speechTagging(List<Vertex> vertexList)。

词性标注使用了隐马，因此涉及到词典CoreNatureDictionary.tr.txt。同样因为我们没有14年的人民日报语料，因此这一部分一般也不修改。需要注意speechTagging默认为false，指的是不进行隐马的词性标注，但不表示系统不进行词性标注，此时系统使用最大概率的词性作为词语的词性，因此CoreNatureDictionary.txt是很重要的。

### 多线程分词

HanLP的ViterbiSegment分词器类是支持多线程的，线程数量由配置变量threadNumber决定的，该变量默认为1。HanLP作者说ViterbiSegmet分词效率最高的原因肯定也有ViterbiSegment分词器支持多线程分词这个因素。另外由于ViterbiSegment分词器内部所具有的相关命名实体功能，因此这些命名实体识别的效率也会很高。在哪里实现的多线程分词呢，在Segment类的List<Term> seg(String text)这个方法中实现的，需要注意HanLP的多线程分词指的是一次输入了一个长文本，而不是一次处理多个输入文本。

# 词性标注

## 概述

### HanLP名词词性标记集

|  |  |  |  |
| --- | --- | --- | --- |
| 分类 | 符号 | 名称 | 示例 |
|  | n | 名词 |  |
| ng | 名词性语素 |  |
| nl | 名词性惯用语 |  |
|  | nx | 字母专名 |  |
| nz | 其他专名 |  |
|  | nb | 生物名 |  |
| nba | 动物名 |  |
| nbc | 动物纲目 |  |
| nbp | 植物名 |  |
|  | nh | 医药疾病等健康相关名词 |  |
| nhd | 疾病名 |  |
| nhm | 药品名 |  |
| nth | 医药 |  |
|  | nr | 人名 |  |
| nr1 | 复姓 |  |
| nr2 | 蒙古姓名 |  |
| nrf | 音译人名 |  |
| nrj | 日本人名 |  |
|  | ns | 地名 |  |
| nsf | 音译地名 |  |
| 普通机构 | nt | 机构团体名 |  |
| ni | 机构相关（不是独立机构） |  |
| nic | 下属机构 |  |
| nis | 机构后缀 |  |
|  | nit | 教育相关机构 |  |
|  | nts | 中小学 |  |
|  | ntu | 大学 |  |
|  | ntc | 公司名 |  |
|  | ntcb | 银行 |  |
|  | ntcf | 工厂 |  |
|  | ntch | 酒店宾馆 |  |
|  | nto | 政府机构 |  |

## 基于统计的词性标注方法

## 基于规则的词性标注方法

# 人名识别

## 调研

### 为什么要进行人名识别？

在对1998年1月语料库上，未登录词占比1%，而未登录词中约50%为人名，因此人名识别对提高分词准确率是有帮助的。

### 基于角色标注的中国人名识别

在HanLP中，基于角色标注识别了中国人名。首先系统利用隐马尔可夫模型标注每个词语的角色，之后利用最大模式匹配法对角色序列进行匹配，匹配上模式的即为人名。理论指导文章为：《基于角色标注的中国人名自动识别研究》，网页地址为：<http://www.hankcs.com/nlp/chinese-name-recognition-in-actual-hmm-viterbi-role-labeling.html>

#### 人名角色简表

所有与人名有关的文本信息被称为角色，比如人名的前词，后词，首字，尾字等。以“馆/内/陈列/周/恩/来/和/邓/颖/超生/前/使用/过/的/物品/。”为例说明其中词语的角色序列。这句话的角色序列为“馆/A内/A 陈列/K周/B恩/C来/D和/M邓/B颖/C超生/V前/A使用/A过/A的/A物品/A。/A”。

不同的实体有不同的角色标注，实际就是特征，这些特征需要有语言学的知识，实际上就是你的阅读量，通过你大量阅读总结经验，比如姓氏可以作为名字的一个特征（张、王、李、赵），常用地名的后缀可以作为一个特征（省、市、区、县），机构名表处所的尾字可以作为一个特征（局、处、所、院）。

下表总结了人名识别中所有角色种类及其标记：

|  |  |  |
| --- | --- | --- |
| 标记 | 意义 | 例子 |
| B | 姓氏 | 张华平先生 |
| C | 双名的首字 | 张华平先生 |
| D | 双名的末字 | 张华平先生 |
| E | 单名 | 张浩说：“我是一个好人” |
| F | 前缀 | 老刘、小李 |
| G | 后缀 | 王总、刘老、肖氏、吴妈、叶帅 |
| K | 人名的上文 | 又来到于洪洋的家。 |
| L | 人名的下文 | 新华社记者黄文摄 |
| M | 两个中国人名之间的成分 | 编剧邵钧林和稽道青说 |
| U | 人名的上文和姓成词 | 这里有关天培的壮烈 |
| V | 人名的末字和下文成词 | 龚学平等领导, 邓颖超生前 |
| X | 姓与双名的首字成词 | 王国维、 |
| Y | 姓与单名成词 | 高峰、汪洋 |
| Z | 双名本身成词 | 张朝阳 |
| A | 以上之外其他的角色 |  |

#### 怎样通过角色序列识别出人名？

基于角色序列的人名识别就是一个简单的模式最大匹配问题。人名模式的识别集为**{ BBCD, BBE, BBZ, BCD, BEE,BE,BG,BXD,BZ,CD,EE,FB, Y,XD}**。例句“馆/内/陈列/周/恩/来/和/邓/颖/超生/前/使用/过/的/物品/。”对应的T#为：“AAKBCDMBC*V*AAAAAA”。V分裂处理后，最终的角色序列为：“AAKBCDMBC*DL*AAAAAA”。模式最大匹配后，我们识别出的人名是：“周恩来”和“邓颖超”。

### 隐马的三要素

观察值序列：词1 词2 … 词n

隐变量序列：角色1 角色2 … 角色n

#### 初始概率分布

初始概率分布即为各种角色的概率分布，如下表所示

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| B | C | D | E | F | G | K | L | M | U | V | X | Y | Z | A |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

#### 状态转移概率分布

状态转移概率分布即为各角色之间转移的概率分布，如下表所示

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | B | C | D | E | F | G | K | L | M | U | V | X | Y | Z | A |
| B |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| C |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| D |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| E |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| F |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| G |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| K |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| L |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| M |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| U |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| V |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Y |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Z |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

#### 观测概率分布

观测概率分布即各角色发射出各词语的概率分布，如下表所示

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 词1 | 词2 | … | 词|V| |
| B |  |  | … |  |
| C |  |  | … |  |
| D |  |  | … |  |
| E |  |  | … |  |
| F |  |  | … |  |
| G |  |  | … |  |
| K |  |  | … |  |
| L |  |  | … |  |
| M |  |  | … |  |
| U |  |  | … |  |
| V |  |  | … |  |
| X |  |  | … |  |
| Y |  |  | … |  |
| Z |  |  | … |  |
| A |  |  | … |  |

词|V|表示训练集词汇表长度。

## 实践

### HanLP人名识别

HanLP有关人名识别的方法有三个，分别为（1）基于HMM-Viterbi角色标注的中国人名识别。（2）基于感知机的实体识别（3）基于CRF的实体识别。本节演示使用基于HMM-Viterbi角色标注的中国人名识别。该方法速度快但精度不高。

|  |
| --- |
| *#coding:utf-8 """ 中文人名识别简单示例 """* **from** pyhanlp **import** \*   **def** demo\_chinese\_name\_recognition(sentences):  segment = HanLP.newSegment().**enableNameRecognize**(**True**)  **for** sentence **in** sentences:  term\_list = segment.seg(sentence)  print(term\_list)   **if** \_\_name\_\_ == **"\_\_main\_\_"**:  sentences = [**"签约仪式前，秦光荣、李纪恒、仇和等一同会见了参加签约的企业家。"**,  **"武大靖创世界纪录夺冠，中国代表团平昌首金"**,  **"区长庄木弟新年致辞"**,  **"朱立伦：两岸都希望共创双赢 习朱历史会晤在即"**,  **"陕西首富吴一坚被带走 与令计划妻子有交集"**,  **"据美国之音电台网站4月28日报道，8岁的凯瑟琳·克罗尔（凤甫娟）和很多华裔美国小朋友一样，小小年纪就开始学小提琴了。她的妈妈是位虎妈么？"**,  **"凯瑟琳和露西（庐瑞媛），跟她们的哥哥们有一些不同。"**,  **"王国强、高峰、汪洋、张朝阳光着头、韩寒、小四"**,  **"张浩和胡健康复员回家了"**,  **"王总和小丽结婚了"**,  **"编剧邵钧林和稽道青说"**,  **"这里有关天培的有关事迹"**,  **"龚学平等领导说,邓颖超生前杜绝超生"**]  demo\_chinese\_name\_recognition(sentences) |

识别结果

|  |
| --- |
| [签约/vi, 仪式/n, 前/f, ，/w, **秦光荣/nr**, 、/w, **李纪恒/nr**, 、/w, **仇和/nr**, 等/udeng, 一同/d, 会见/v, 了/ule, 参加/v, 签约/vi, 的/ude1, 企业家/nnt, 。/w]  [**武大靖/nr**, 创/v, 世界纪录/nz, 夺冠/vi, ，/w, 中国代表团/nt, 平昌/ns, 首金/n]  [区长/nnt, **庄木弟/nr**, 新年/t, 致辞/vi]  [**朱立伦/nr**, ：/w, 两岸/n, 都/d, 希望/v, 共创/v, 双赢/n, /w, **习/v**, **朱/ag**, 历史/n, 会晤/vn, 在即/vi]  [陕西/ns, 首富/n, **吴一坚/nr**, 被/pbei, 带走/v, /w, 与/cc, **令计划/nr**, 妻子/n, 有/vyou, 交集/v]  [据/p, 美国之音/n, 电台/nis, 网站/n, 4月/t, 28/m, 日/b, 报道/v, ，/w, 8/m, 岁/qt, 的/ude1, **凯瑟琳·克罗尔/nrf**, （/w, **凤甫娟/nr**, ）/w, 和/cc, 很多/m, 华裔/n, 美国/nsf, 小朋友/n, 一样/uyy, ，/w, 小小年纪/n, 就/d, 开始/v, 学/v, 小提琴/n, 了/ule, 。/w, 她/rr, 的/ude1, 妈妈/n, 是/vshi, 位/q, 虎妈/nz, 么/y, ？/w]  [**凯瑟琳/nrf**, 和/cc, **露西/nrf**, （/w, **庐瑞媛/nr**, ）/w, ，/w, 跟/p, 她们/rr, 的/ude1, 哥哥/n, 们/k, 有/vyou, 一些/m, 不同/a, 。/w]  [**王国强/nr**, 、/w, **高峰/n**, 、/w, **汪洋/n**, 、/w, **张朝阳/nr**, 光着头/l, 、/w, **韩寒/nr**, 、/w, **小/a, 四/m**]  [**张浩/nr**, 和/cc, **胡健康/nr**, 复员/v, 回家/vi, 了/ule]  [**王总/nr**, 和/cc, **小丽/nr**, 结婚/vi, 了/ule]  [编剧/nnt, **邵钧林/nr**, 和/cc, **稽道青/nr**, 说/v]  [这里/rzs, 有/vyou, **关天培/nr**, 的/ude1, 有关/vn, 事迹/n]  [**龚学平/nr**, 等/udeng, 领导/n, 说/v, ,/w, **邓颖超/nr**, 生前/t, 杜绝/v, 超生/vi] |

从初步实践结果看，基于HMM的Viterbi人名识别方法准确率不行，召回率也不行。

### Jieba人名识别

|  |
| --- |
| print("人名识别示例：") sentences = ["签约仪式前，秦光荣、李纪恒、仇和等一同会见了参加签约的企业家。",  "武大靖创世界纪录夺冠，中国代表团平昌首金",  "区长庄木弟新年致辞",  "朱立伦：两岸都希望共创双赢 习朱历史会晤在即",  "陕西首富吴一坚被带走 与令计划妻子有交集",  "据美国之音电台网站4月28日报道，8岁的凯瑟琳·克罗尔（凤甫娟）和很多华裔美国小朋友一样，小小年纪就开始学小提琴了。她的妈妈是位虎妈么？",  "凯瑟琳和露西（庐瑞媛），跟她们的哥哥们有一些不同。",  "王国强、高峰、汪洋、张朝阳光着头、韩寒、小四",  "张浩和胡健康复员回家了",  "王总和小丽结婚了",  "编剧邵钧林和稽道青说",  "这里有关天培的有关事迹",  "龚学平等领导说,邓颖超生前杜绝超生"] for sentence in sentences:  words = pseg.cut(sentence)  for w in words:  print('%s/%s' % (w.word, w.flag), end=' ')  print() |

识别结果

|  |
| --- |
| 签约/v 仪式/n 前/f ，/x **秦/nr 光荣/nr** 、/x **李纪恒/nr** 、/x Prefix dict has been built succesfully.  **仇和/nr** 等/u 一同/d 会见/n 了/ul 参加/v 签约/v 的/uj 企业家/n 。/x  **武大/j 靖/nr** 创/n 世界纪录/nz 夺冠/v ，/x 中国/ns 代表团/n 平昌/nr 首金/n  区长/n **庄木弟/nr** 新年/t 致辞/v  **朱立伦/nr** ：/x 两岸/f 都/d 希望/v 共创/n 双赢/nr /x **习朱/nr** 历史/n 会晤/v 在/p 即/v  陕西/ns 首富/n **吴一坚/nr** 被/p 带走/v /x 与/p 令/nrt 计划/n 妻子/n 有/v 交集/v  据/p 美国之音/nz 电台/n 网站/n 4/m 月/m 28/m 日/m 报道/v ，/x 8/m 岁/m 的/uj 凯瑟琳/nrt ·/x **克罗/nrt 尔/nr** （/x **凤甫娟/nr** ）/x 和/c 很多/m 华裔/nr 美国/ns 小朋友/n 一样/r ，/x 小小年纪/n 就/d 开始/v 学/n 小提琴/nr 了/ul 。/x 她/r 的/uj 妈妈/n 是/v 位/q 虎/n 妈/n 么/y ？/x  **凯瑟琳/nrt** 和/c **露西/n** （/x **庐/ng 瑞/j 媛/x** ）/x ，/x 跟/p 她们/r 的/uj 哥哥/ns 们/k 有/v 一些/m 不同/a 。/x  **王国强/nr** 、/x **高峰/nr** 、/x **汪洋/nr** 、/x **张朝阳/nr** 光着头/l 、/x **韩寒/nr** 、/x **小四/nr**  **张浩/nr** 和/c **胡/nr 健康/a** 复员/n 回家/n 了/ul  **王/nr 总**和/n **小丽/nr** 结婚/v 了/ul  编剧/n **邵钧林/nr** 和/c **稽道/v 青**说/n  这里/r 有/v **关天培/nr** 的/uj 有关/vn 事迹/n  **龚学平/nr** 等/u 领导/n 说/v ,/x **邓颖超/nr** 生前/t 杜绝/nr 超生/v |

### HanLP与Jieba识别效果对比

#### 只有Jieba识别出的人名

准确率极低，基本为地名或复杂地名组成部分或复杂机构名组成部分。举例如下：

[1] 战乱的**阿富汗**地区，枪支可随意买卖，AK47价格约500人民币

“阿富汗”被识别为人名。

[2] 安庆到**桂林**自驾游如何规划？

“桂林”被识别为人名。

[3] 2018天津市和平分局招聘社区戒毒、社区**康复**工作人员成绩查询入口

“康复”被识别为人名。

#### 只有HanLP识别出的人名

除了特别常用姓氏的名字识别正确，其他的都识别错误。举例如下：

[1] 纳溪区副区长李明带队到“**花田酒**地”景区检查节前安全工作

“花田酒”被被识别为人名。

[2] 秀英“线上线下”**齐发力** 助力贫困户“微互动”拓宽农产品销路

“齐发力”被识别为人名。

[3] 紧急通知：**秦报**融媒粉团祖山一日游日报名费大调整！

“秦报”被识别为人名。

#### HanLP与Jieba都识别出的人名

1. 非常用姓氏识别出的人名基本错误。

[1] 房产高管薪酬大起底 万科**郁亮**年薪1189.9万仅排第二

[2] 生生不息 南通**支云**发布汶川地震十周年海报呼吁赛前默哀

[3] 为什么**伊郎**不能有核武器，而美国有核武器？

2. 名字本身构成词时基本错误。

[1] 周口一村庄**杨絮**着火，对付杨絮用啥方法好呢？

[2] 上联: 三国**魏蜀吴**，如何对下联？

[3] 上联:灯火辉煌**万家乐**。求下联？

## badcase分类整理

### 核心词典中的单字姓氏人名引起误识别

核心词典中的单字人名姓氏影响

[1] 日本人为什么会推崇降臣王**铎**的书法？

[2] 徐**峥**小陶虹夫妇《信中国》，致敬汶川地震救灾英雄，缅怀遇难同胞

[3] **嵇**境雷——新时代最具收藏价值的书画名家作品展

解决方法：去掉核心词典中的单字姓氏，将没有识别全的名字放在人工干预表中。核心词典中最好不要放单名姓氏。

### 人名、地名混淆引起的误识别

[1] **马嵬驿**和白鹿原旅游景区的游客为什么日益锐减？

[2] **韩美今**启动“超级雷霆”联合军演，8架F22战机参演

[3] 安徽**毛坦厂**镇各地学生就读 逾万家长陪读

解决方法：核心词典中进行地名干预。

### 双名本身成词、姓与双名首字成词、姓与单名成词、人名末字和下文成词、人名上文和姓成词

[1] 房价**高全怪**炒房客 炒房客：不炒房我们炒什么？

[2] 开发**商抢地**还抢人！南阳房地产市场要开挂！

[3] 水磨镇涅槃重生 工业重镇成功转型**康养**旅游景区

解决方法：

（1）将人名词典中的成词人名标记为忽略，这样的词语也应该放到核心词典中。

（2）HanLP发射矩阵词典nr.txt中收录单字姓氏619个。袁义达在《中国的三大姓氏是如何统计出来的》文献中指出：当代中国100个常见姓氏中，集中了全国人口的87%，根据这一数据我们只保留nr.txt中的100个常见词语的姓氏角色，其他词语去掉其姓氏角色状态。过滤后，nr.txt中具有姓氏角色的单字共计97个。列于下表：

|  |
| --- |
| 丁 万 乔 于 任 何 余 侯 傅 冯 刘 卢 史 叶 吕 吴 周 唐 夏 姚  姜 孔 孙 孟 宋 尹 崔 常 康 廖 张 彭 徐 戴 方 易 曹 曾 朱 李  杜 杨 林 梁 武 段 毛 江 汤 汪 沈 潘 熊 王 田 白 石 秦 程 罗  胡 苏 范 萧 董 蒋 薛 袁 许 谢 谭 贺 贾 赖 赵 邓 邱 邵 邹 郑  郝 郭 金 钟 钱 阎 陆 陈 雷 韩 顾 马 高 魏 黄 黎 龚 |

对于非常用姓氏构成的人名，我们人工干预那些3字以上的人名，其他的人名不做识别。

（3）双字人名模式过滤

我们对87个误识别的双字人名的角色序列进行了统计，人名和其对应角色列于下表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 奥鹏 | EE | 谷寿 | EE | 交安 | EE | 青创 | EE | 温阳 | EE |
| 小司 | FC | 龟峰 | EE | 于练 | BC | 青洋 | EE | 吴公 | BC |
| 霸屏 | EE | 过么 | EE | 京云 | EE | 屈光 | EE | 武文 | BC |
| 白蝶 | BE | 函谷 | EE | 于右 | BC | 任城 | BE | 袭黔 | EE |
| 小田 | FB | 韩美 | BC | 康养 | BC | 任君 | BC | 弦流 | EE |
| 星繁 | EE | 叶绿 | BC | 抗凝 | EE | 沙鹰 | EE | 小麦 | FC |
| 超超 | EE | 寒香 | EE | 原牧 | EE | 赏雪 | EE | 小密 | FE |
| 晨峰 | EE | 何会 | BC | 蓝德 | EE | 射导 | EE |  |  |
| 薛家 | BC | 何有 | BC | 浪淘 | EE | 沈师 | BC |  |  |
| 电竞 | EE | 易疆 | BE | 老多 | FC | 施策 | EE |  |  |
| 方宅 | BE | 胡赛 | BC | 霑梅 | EE | 熟宣 | EE |  |  |
| 非遗 | EE | 易旺 | BC | 黎坪 | BE | 帅铃 | EE |  |  |
| 枫示 | EE | 音悦 | EE | 张图 | BC | 苏城 | BE |  |  |
| 高带 | BC | 黄安 | BC | 毛体 | BC | 唐口 | BC |  |  |
| 何造 | BC | 黄貂 | BE | 郑医 | BC | 唐冶 | BC |  |  |
| 高屋 | BE | 黄鹤 | BC | 周起 | BC | 塘峪 | EE |  |  |
| 燕舞 | EE | 鹰爸 | EE | 周周 | BC | 添香 | EE |  |  |
| 龚滩 | BE | 鹰跃 | EE | 钱会 | BC | 汪山 | BE |  |  |
| 杨飘 | BC | 影添 | EE | 茜溪 | EE | 王洞 | BC |  |  |
| 杨絮 | BC | 江静 | BE | 秦报 | BE | 唯逢 | EE |  |  |

各角色序列频次分布如下

|  |  |
| --- | --- |
| EE | 41 |
| BC | 29 |
| BE | 12 |
| FC | 3 |
| FB | 1 |
| FE | 1 |

目前只保留下边的双字（词）角色模式

|  |  |
| --- | --- |
| 模式串 | 示例 |
| BG | 王总 |
| BZ | 张朝阳 |
| FB | 老张 |

（4）三字人名模式过滤

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 韩美今 | BCD | 才子多 | XD | 古树群 | XD | 吉祥新 | XD |
| 高全怪 | BCD | 陈仓米 | XD | 古思特 | XD | 江景房 | XD |
| 华十大 | BCD | 成人礼 | XD | 关东军 | XD | 金龙盘 | XD |
| 巴马香 | XD | 杜蕾斯 | XD | 胡里山 | XD | 康复为 | XD |
| 白音昌 | XD | 宫川英 | XD | 花田酒 | XD | 来华政 | XD |

从上表可以看出，识别出的姓与双名首字成词的角色构成的人名基本都被识别错了。

目前只保留下边的双字（词）角色模式

|  |  |
| --- | --- |
| 模式串 | 示例 |
| BBE | 刘张虎 |
| BCD | 张华平 |
| BEE | 林华华 |

### 双名末字为动词

[1] 著名书法家**张海为**自己立“卧薪尝胆状”

解决方法：将误识别的人名标注为忽略。如果后续这种人名较多了，则将“为”字作为双名末字的角色去掉。

### 对联、诗歌等文学作品中由姓氏引发的人名误识别

[1] 上联：**常在花**间走，如何对下联？

[2] 上联：**江中鱼**东游西藏，下联怎么对？

[3] 楷书《**石涧记**》

解决方法：目前只能发现后标注为忽略。

# 地名识别

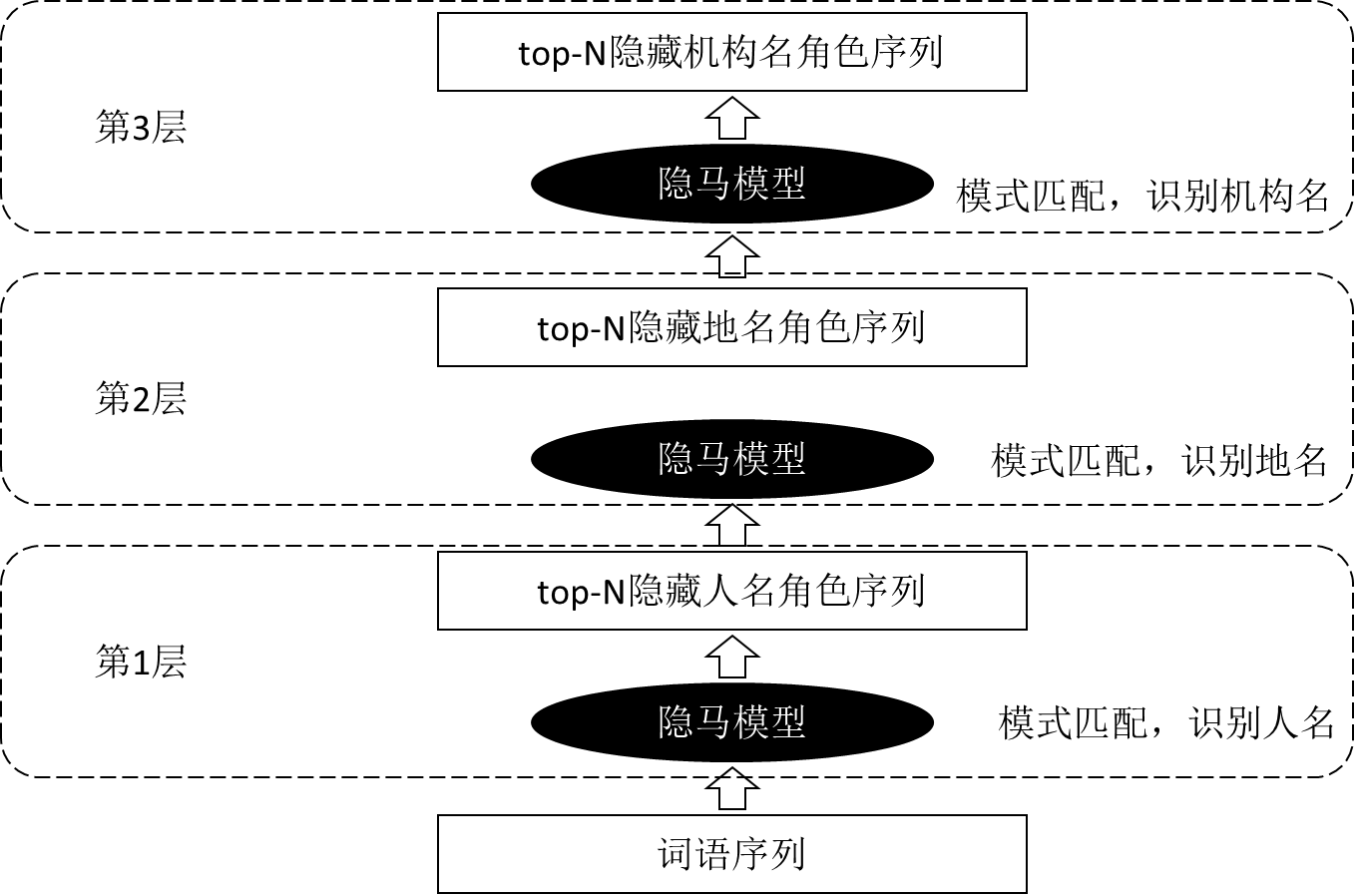
## 调研

### 基于角色标注的中国地名识别

理论指导文章为：《基于层叠隐马尔可夫模型的中文命名实体识别》，网页地址为：

<http://www.hankcs.com/nlp/ner/place-names-to-identify-actual-hmm-viterbi-role-labeling.html>。

#### 层叠隐马



如图所示，层叠隐马就是训练三个隐马模型，每个模型标注一种实体，三个模型采用级联形式连接。

#### 地名角色简表

|  |  |  |
| --- | --- | --- |
| 角色 | 意义 | 例子 |
| A | 地名的上文 | 我/来到/中/关/园 |
| B | 地名的下文 | 刘家村/和/下岸村/相邻 |
| C | 中国地名的首部 | 石/河/子/乡 |
| D | 中国地名的中部 | 石/河/子/乡 |
| F | 中国地名的末部 | 石/河/子/乡 |
| G | 中国地名的后缀 | 海/淀区 |
| X | 连接词 | 刘家村/和/下岸村/相邻 |
| Z | 其他非名成分 |  |

## 实践

### HanLP地名识别

本节演示地名识别。

|  |
| --- |
| *# # -\*- coding:utf-8 -\*- # Author：wancong # Date: 2018-04-30* **from** pyhanlp **import** \*  Segment = JClass(**"com.hankcs.hanlp.seg.Segment"**) Term = JClass(**"com.hankcs.hanlp.seg.common.Term"**)  **def** demo\_place\_recognition(sentences):  segment = HanLP.newSegment().**enablePlaceRecognize**(**True**)  **for** sentence **in** sentences:  term\_list = segment.seg(sentence)  print(term\_list)   **if** \_\_name\_\_ == **"\_\_main\_\_"**:  sentences = [**"蓝翔给宁夏固原市彭阳县红河镇黑牛沟村捐赠了挖掘机"**]  demo\_place\_recognition(sentences) |

识别结果

|  |
| --- |
| [蓝翔/nt, 给/p, **宁夏/ns**, **固原市/ns**, **彭阳县/ns**, **红河镇/ns**, **黑牛沟村/ns**, 捐赠/v, 了/ule, 挖掘机/n] |

### Jieba地名识别

|  |
| --- |
| *# 地名识别* print(**"地名识别"**) sentences = [**"蓝翔给宁夏固原市彭阳县红河镇黑牛沟村捐赠了挖掘机"**] **for** sentence **in** sentences:  words = pseg.cut(sentence)  **for** w **in** words:  print(**'%s/%s'** % (w.word, w.flag), end=**' '**)  print() |

识别结果

|  |
| --- |
| 蓝翔/nr 给/p **宁夏/ns 固原市/ns 彭阳县/ns** **红河/ns 镇黑牛沟/ns 村**/n 捐赠/v 了/ul 挖掘机/n |

## 改进

使能HMM的人名、地名、机构名识别的效果。

nr 正确预测个数314, 预测总数1016, precision = 30.91

ns 正确预测个数3439, 预测总数4185, precision = 82.17

nt 正确预测个数301, 预测总数1346, precision = 22.36

使能HMM的人名、机构名识别的效果。

nr 正确预测个数314, 预测总数1048, precision = 29.96

ns 正确预测个数3333, 预测总数3962, precision = 84.12

nt 正确预测个数302, 预测总数1374, precision = 21.98

使能HMM的人名、地名识别的效果。

nr 正确预测个数315, 预测总数1045, precision = 30.14

ns 正确预测个数3606, 预测总数4588, precision = 78.60

nt 正确预测个数29, 预测总数34, precision = 85.29

使能人名识别的效果

nr 正确预测个数315, 预测总数1083, precision = 29.09

ns 正确预测个数3488, 预测总数4332, precision = 80.52

nt 正确预测个数29, 预测总数34, precision = 85.29

分析：

（1）人名识别默认打开。

（2）地名识别是否打开对精度影响不大，因此初期可以打开地名识别。

（3）机构名打开后，精度下降很大，因此在初期不做基于HMM的机构名识别。而暂时只用词表匹配法。

# 机构名识别

## 调研

### 基于角色标注的中国机构名识别

机构名的理论指导文章为：《基于角色标注的中文机构名识别》，网页地址为：

<http://www.hankcs.com/nlp/ner/place-name-recognition-model-of-the-stacked-hmm-viterbi-role-labeling.html>。

### 机构名角色标注简表

|  |  |  |
| --- | --- | --- |
| 角色 | 意义 | 例子 |
| A | 上文 | *参与*亚太经合组织的活动 |
| B | 下文 | 中央电视台*报道* |
| X | 连接词 | 北京电视台*和*天津电视台 |
| C | 特征词的一般性前缀 | 北京*电影*学院 |
| F | 特征词的译名性前缀 | 美国*摩托罗拉*公司 |
| G | 特征词的地名性前缀 | 交通银行*北京*分行 |
| H | 特征词的机构名前缀 | *中共中央*顾问委员会 |
| I | 特征词的特殊性前缀 | *中央*电视台 |
| J | 特征词的简称性前缀 | *巴*政府 |
| D | 机构名的特征词 | 国务院侨务*办公室* |
| Z | 非机构名成份 |  |

## 实践

### HanLP机构名识别

本节演示机构名识别。

|  |
| --- |
| *# # -\*- coding:utf-8 -\*- # Author：wancong # Date: 2018-04-30* **from** pyhanlp **import** \*  Segment = JClass(**"com.hankcs.hanlp.seg.Segment"**) Term = JClass(**"com.hankcs.hanlp.seg.common.Term"**)   **def** demo\_organization\_recognition(sentences):  segment = HanLP.newSegment().enableOrganizationRecognize(**True**)  **for** sentence **in** sentences:  term\_list = segment.seg(sentence)  print(term\_list)   **if** \_\_name\_\_ == **"\_\_main\_\_"**:  sentences = [**"我在上海林原科技有限公司兼职工作，"**,  **"我经常在台川喜宴餐厅吃饭，"**,  **"偶尔去开元地中海影城看电影。"**]  demo\_organization\_recognition(sentences) |

识别结果

|  |
| --- |
| [我/rr, 在/p, 上海/ns, **林原科技有限公司/nt**, 兼职/vn, 工作/vn, ，/w]  [我/rr, 经常/d, 在/p, **台川喜宴餐厅/nt**, 吃饭/vi, ，/w]  [偶尔/d, 去/vf, **开元地中海影城/nt**, 看/v, 电影/n, 。/w] |

### Jieba机构名识别

## 改进

|  |
| --- |
| print(**"\n\n机构名识别"**) sentences = [**"我在上海林原科技有限公司兼职工作，"**,  **"我经常在台川喜宴餐厅吃饭，"**,  **"偶尔去开元地中海影城看电影。"**] **for** sentence **in** sentences:  words = pseg.cut(sentence)  **for** w **in** words:  print(**'%s/%s'** % (w.word, w.flag), end=**' '**)  print() |

识别结果

|  |
| --- |
| 我/r 在/p **上海/ns 林原/nr 科技/n 有限公司/n** 兼职/v 工作/vn ，/x  我/r 经常/d 在/p **台川/ns 喜宴/v 餐厅/n** 吃饭/v ，/x  偶尔/d 去/v **开元/nz 地中海/ns 影城/n** 看/v 电影/n 。/x |

# 医疗领域实体识别

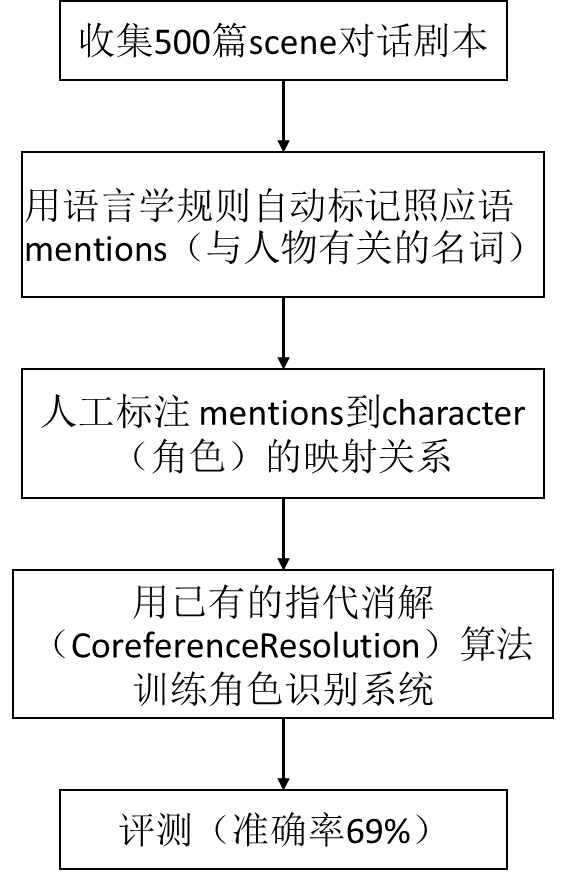
## 调研

## 药品名识别

## 疾病名识别

# 角色识别

## 角色识别系统构建流程



## 指代消解

指代一般分为两种情况：回指（anaphora，也称为指示性指代）和共指（co-reference，也称同指）。所谓回指，是指当前的指示语与上下文出现的词、短语或句子（句群）存在密切的语义关联性；共指则主要是指两个名词（包括代名词、名词短语）指向真实世界中的同一参照体。共指关系可以独立于上下文存在。例如，“美国总统”和“布什”就是共指。

指代消解是指将文本中指称同一对象的所有表达形式关联起来。包括具有指代关系的指示语和先行语，以及具有同指关系的词语。分析文档中实体之间的指代关系，同一个实体在所分析的篇章中，可能有多种不同的指代方法，指代的解析就是将不同的指代连接到同一实体上。