中文分词相关算法研究

中文分词(Chinese Word Segmentation) 指的是**将一个汉字序列切分成一个一个单独的词。**

一、 基本中文分词方法

当前实际使用的中文分词方法可以归为两大类：基于词典的分词和基于统计的分词

1、 基于词典（字符串匹配）的分词

（1）正向最大匹配（从左到右）

（2）逆向最大匹配（从右到左）

（3）最小匹配

（4）双向最大匹配（从左到右和从右到左两次扫描，可使用 ngram 词典选择最优的匹配）

（5）基于词图进行搜索

优点：速度快、便于维护，容易适应领域

缺点：对于歧义和未登录词效果不佳

2、 基于统计的分词

给定语料的前提下，训练统计机器学习模型进行分词。

将分词问题抽象为序列标注问题。

基于统计的分词是当前中文分词的主流方法，主要的统计模型为：

（1）隐马尔可夫模型（Hidden Markov Model ， HMM）

（2）最大熵模型（ME）

（3）条件随机场模型（Conditional Random Fields，CRF）

实际使用中往往将两者结合起来，使用统计方法识别新词、消除歧义；利用词典快速分词并增加一定的领域适应性。

二、 中文分词的技术难点

1、 歧义识别

（1）交集型歧义:

一个字既可以和前面的字结合成词，也可以和后面的字结合成词。

例如："这几块地面积还真不小"

（2）组合型歧义：

例如："将军任命了一名中将", "产量三年中将增长两倍"

（3）真歧义：

人也无法区分判断的歧义

例如："乒乓球拍卖完了"

2、 未登录词识别

分词词典中没有收录的词称为未登录词，例如新词、专业术语和命名实体（人名、地名和组织机构名）。

三、 基于感知器的序列标注算法

仍然属于基于统计的方法，将问题抽象为序列标注问题。

不计算概率分布，只计算标注序列的得分：

$$score(\mathrm{X},\mathrm{Y}) = \sum_{k=1}^{K} w_k f_k(X,Y)$$

其中$f_k(X,Y)$为特征函数，w_k为第 k 个特征对应的权重。

得分最大的标注序列为最佳标注序列：

$$Y = argmax_{Y'} score(X, Y')$$

通过训练得到权重 w，然后通过 viterbi 算法快速得到使 score 最大的标注序列。

感知器算法的优点：

1、感知器属于在线算法，每次可以只使用一个实例对参数进行更新

实际使用中为了防止过拟合，一般在更新参数的时候进行平均化操作，称为 Average Perceptron 算法。

2、可并行训练

3、方便进行模型压缩

4、可对原始模型进行增量训练

5、和 CRF 等模型相比比较简单

实现感知器算法的关键步骤如下：

1、构造特征

对分词算法特征的选取直接影响最终分词模型的性能，构造特征时考虑如下方法：

（1）字符 n-gram 特征

一元特征，如当前字符、前一个字符、后一个字符，等等

二元特征，当前-前一个字符组合、当前-后一个字符组合，等等

三元特征，前-当前-后共三个字符的组合，等等

（2）字符类别

如果字符是否为数字、字母、停用词等

2、字典融入

感知器可以使用字典，增加分词性能并提高领域适应性

感知器并不直接使用字典进行分词，而是使用字典特征的方式将字典融入感知器

具体使用哪些字典特征需要进行实际试验和测试

3、感知器并行训练

使用迭代参数融合算法：

（1）在多个不相交子集数据上训练多个子模型

（2）每一轮训练结束后对参数进行融合

（3）每个子模型使用融合后的参数继续下一次迭代

4、模型压缩

前提：

1、感知器算法的特征数量可能会很大

2、很大一部分特征的权重可能很小，对计算序列的分数影响十分小

好处：

1、减小模型文件大小

2、降低对内存的使用

方法：

1、设置压缩比

2、根据压缩比设定阈值，将权重绝对值小于阈值的特征删除

5、增量训练

使用原始模型+增量语料进行训练，分两种情况：

1、增量语料和原始模型属于同一领域

使用增量语料训练子模型，然后和原始模型进行融合

2、增量语料和原始模型不属于同一领域

基于两层的 Stacked Learning 框架：

（1）第一层为由初始语料训练的初始模型

（2）初始模型的分词结果和原始句子作为第二层的输入，

这样第二层的模型可以使用第一层的标注结果作为特征来学习到不同领域的差异

四、 基于深度学习的分词方法

深度学习主要是特征学习，进行端到端的训练，需要使用大量语料。

可以利用 GPU 来大幅度提高训练速度。

其中一种方法是基于论文《Neural Architectures for Named Entity Recognition》

论文中使用的方法为：

字嵌入+双向 LSTM+CRF，字嵌入可以是预先训练好的 Word2Vec 模型

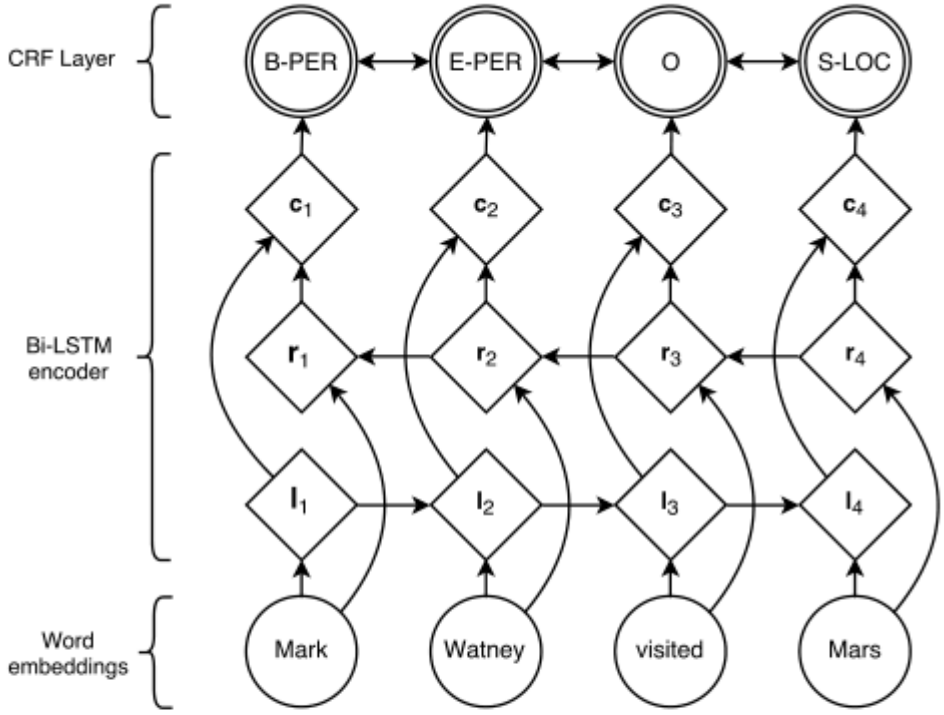

Figure 1: Main architecture of the network. Word embeddings are given to a bidirectional LSTM. $\mathbf{l}_i$ represents the word i and its left context, $\mathbf{r}_i$ represents the word i and its right context. Concatenating these two vectors yields a representation of the word i in its context, $\mathbf{c}_i$.

五、 几种开源中文分词系统的比较

LTP - 结构化感知器

CRF++ - CRF

jieba - 词典+HMM

FoolNLTK – 深度学习，BiLSTM-CRF

Cws2 – 自己开发的平均感知机分词器，使用 Python 开发

Bakeoff 2005 PKU 数据集：

分词器	准确率	召回率	F1 score
LTP	0.928	0.929	0.929
CRF++	0.926	0.942	0.934
Jieba	0.786	0.853	0.819
foolNLTK	0.93	0.92	0.93
Cws2（自己开发）	0.939	0.925	0.932

Bakeoff 2005 MSR 数据集：

分词器	准确率	召回率	F1 score
LTP	0.940	0.945	0.942
CRF++	0.965	0.961	0.963
Jieba			
foolNLTK			
Cws2（自己开发）	0.962	0.971	0.967

注：

1、 CRF++使用模版如下（共 7 个一元特征，1 个二元特征）：

单字符：（当前字）、（前一个字）、（后一个字）

双字符：（前前一个字，前一个字）、（前一个字，当前字）、（当前字，后一个字）、（后一个字，后后一个字）

转移：（前一个字的标注，当前字的标注）

2、 cws2 使用和 CRF++一样的特征模版，并且加载了数据集提供的词典

3、 foolNLTK 使用其自带的字嵌入机制(google 的 word2vec)，未使用外部字向量

4、 jieba 分词未使用词典，jieba 分词的外部词典要求提供词频，数据集的词典不满足要求

5、以上开源系统均使用默认参数进行训练，未进行精细调节

Todo: 在不同数据集上进行性能测试；

测试对于歧义句子的分词能力

歧义句子分词测试（使用 MSR 语料训练）：

1) Cws2.py

这几块地面积还真不小。

这 / 几块 / 地 / 面积 / 还 / 真 / 不 / 小 / 。

将军任命了一名中将。产量三年中将增长两倍。

将军 / 任命 / 了 / 一 / 名 / 中将 / 。产量 / 三年 / 中将 / 增长 / 两倍 / 。

这句话被错误分词，查看字典里面有“中将”这个词，将其删去，测试结果如下：

将军 / 任命 / 了 / 一 / 名 / 中 / 将 / 。 / 产量 / 三年 / 中 / 将 / 增长 / 两倍 / 。

这应该是语料不足/不均衡产生的问题。

六、 主流分词技术的分析

Todo：分析各种主流分词方法在性能、速度、实际应用方面的优劣；

使用 Python 编写简短程序试验、分析优化方向

七、 自制平均感知器进行分词实验：

感知器分词具有实现简单，训练速度快，可在线学习等优点，并且感知器可使用和 CRF 类似的模版来定义特征，分词性能和使用 CRF 进行分词的系统不相上下。

下面使用自己开发的平均感知器 cws2.py 进行了一些简单的分词实验，以加深对中文分词（以及序列标注）问题的理解和认识。

1、 特征模版的定义和使用

特征是影响机器学习算法性能的最关键因素。

感知器算法和 CRF 都可以使用特征模版来定义特征，所以这里的实验研究结论可以很容易的应用到以 CRF 为基础的分词器。

cws2.py 默认使用的特征模版为（X 为输入中文句子，i 为当前字的位置，Y 为标注序列）：

X[i-1]

X[i]

X[i+1]

X[i-2]-X[i-1]

X[i-1]-X[i]

X[i]-X[i+1]

X[i+1]-X[i+2]

(Y[i-1], Y[i])

前面 7 个以句子 X 产生的特征经过加权求和产生类似 HMM 中的发射概率的数值，或许可以称为发射得分。

最后标注序列 Y 产生的特征类似 HMM 中的转移概率，或者转移得分。

	准确率	召回率	F1 score
默认模版	0.9622	0.9712	0.9666
增加 X[i-1]-X[i]-X[i+1]	0.9656	0.9734	0.9695
增加 X[i-2]和 X[i+2]	0.9644	0.9713	0.9678
同时增加 X[i-1]-X[i]-X[i+1] 和 X[i-2]和 X[i+2]	0.9662	0.9728	0.9695

2、 词典的使用或融入方式

词典和统计学习相结合的方法是当前分词系统的主流方向。

依据使用词典的方式大概可以分为两类：

1） 以词典为准

这里面有可以分两种方法：一种是先用词典来分词，剩下的片段使用统计方法，如 jieba 分词；另一种是使用统计学习方法分词（如 CRF），在 decode 的时候查词典，提高其中词汇的标注得分。

这类完全以词典为准来分词的优点：只要词典里面有的词就可以完全切出来，例如人名、地名、公司名等命名实体

缺点：仍然需要解决歧义问题，例如“这几块地面积还真不小”，“地面”和“面积”具有切分歧义，这本来可以通过统计学习来解决，如果完全以词典为准，那么需要其他方法，如使用词频信息。

2） 将词典融入特征

这类方法将词典融入统计学习，**提高**组成词典中词的各个字的某些“成词能力”。

Cws2.py 中使用如下方法来融入词典：

以当前位置及后面的句子片段为字符串，在词典中进行正向最大匹配，如果匹配长度为 L（L 至少为 2），那么产生如下特征模版：

<forward>-X[i]-L

以当前位置及前面的句子片段为字符串，在词典中进行逆向最大匹配，如果匹配长度为 L（L 至少为 2），那么产生如下特征模版：

<backword>-X[i]-L

举例说明：

使用 MSR 语料训练的模型来分词：

穷小子逆袭迎娶白富美。

穷 / 小子 / 逆袭 / 迎娶 / 白 / 富 / 美 / 。

可以看到，“白富美”这个词在 MSR 语料中没有，而“白”、“富”、“美”这三个字单独成词的能力都比较强，所以这里切分出三个词。

将“白富美”加入词典中，切分结果如下：

穷 / 小子 / 逆袭 / 迎娶 / 白富美 / 。

这里使用了正向和逆向最大匹配模版：具体为<forward>-白-3 和<backword>-美-3

查看词典，以“白”开头的三字词有：白毛女、白热化、白米饭、白织灯、白内障等等，词典中的这些词表明了“白”这个字作为三字词的首个字的成词能力。

同理，以“美”作为结尾的三字词有：陈世美、艺术美、自然美、真善美、房利美等等，这些词表明了“美”这个字作为三字词的尾字的成词能力。

这样，在词典中加入了“白富美”后，不需要再进行统计学习，模型就可以正确切分“白富美”这个词。

3、 歧义词语切分

使用 MSR 语料结合语料提供的词典训练模型，测试模型对于歧义词的区分能力：

这几块地面积还真不小。

这 / 几块 / 地 / 面积 / 还 / 真 / 不 / 小 / 。

将军任命了一名中将。产量三年中将增长两倍。

将军 / 任命 / 了 / 一 / 名 / 中将 / 。产量 / 三年 / 中将 / 增长 / 两倍 / 。

上面这句话第二个“中将”没有被切分开，查看字典里面有“中将”这个词，**不使用**词典训练模型，再次测试结果如下：

将军 / 任命 / 了 / 一 / 名 / 中将 / 。 / 产量 / 三年 / 中 / 将 / 增长 / 两倍 / 。

这说明了词典提高了“中将”这两个字成为一个词的能力，词典的使用导致出现了歧义！
而不使用词典进行统计学习，算法见识到了“中”“将”的各种用法，可以正确切分各种组合。
这个实验说明**解决歧义词切分的最佳方法是使用均衡语料进行统计学习。**
同时也说明对于语料中出现频数较高的词不需要使用词典，应该靠统计方法从语料中学习字的成词能力。

下面是另一个例子：
使用词典：

南京市长江大桥。
南京市 / 长江大桥 / 。

某某市长江大桥。
某某 / 市长江大桥 / 。

不使用词典：

南京市长江大桥。
南京市 / 长江大桥 / 。

某某市长江大桥。
某 / 某 / 市 / 长江大桥 / 。

可以看到，因为“市长”这个词在词典里，反而导致出现分词错误。（第一句分词正确，因为“南京市”也在词典里面）