

Draft

# 自然语言处理概论

张奇, 桂韬

August 2, 2021

# 目录

目录	ii
1 绪论	1
1.1 拟定章节 . . . . .	1
3 词法分析	3
3.1 词形分析 . . . . .	3
3.2 词语切分 . . . . .	11
3.3 词性标注 . . . . .	24
3.4 延伸阅读 . . . . .	29
3.5 习题 . . . . .	30
参考文献	31

## 1.1 拟定章节

1.1 拟定章节 . . . . 1

目录:

第一章绪论

第二章语言模型

第三章词法分析

第四章句法分析

第五章语义分析

第六章信息抽取

第七章篇章分析

第八章机器翻译

第九章情感分析

第十章智能问答

第十一章人机对话

第十二章知识图谱

附录:

附录 1 线性模型

附录 2 决策树

附录 3 支持向量机 (SVM)

附录 4 条件随机场 (CRF)

附录 5 卷积神经网络 (CNN)

附录 6 递归神经网络 (RNN+LSTM)

附录 7 变压器模型 (Transformer)

附录 8 图网络 (GCN 等)

附录 9 预训练方法

在语言学中，**词（Word）**是能够独立运用并能够表达语义或语用内容的最基本单元，包含形式和意义两个部分。在由麻省理工大学教授乔姆斯基提出的最简方案（Minimalist Program）语言学派中，词也被认为是包含形式和意义的结构。例如：英文单词“cats”具有语义是“猫”，词性为名词，数量为复数，读音为 mǎo，在语法上需要与其所在句子或者词组的其他成分（如动词等）达成一致等特性。以英语为代表的**印欧语系（Indo-European languages）**中词之间通常有分隔符（空格等）来区分。但是以汉语为代表的**汉藏语系（Sino-Tibetan languages）**，以及以阿拉伯语为代表的**闪-含语系（Semito-Hamitic languages）**中却不包含明显的词之间的分隔符。由于词是最小的语义单元，因此在自然语言处理中，词的处理通常是句法分析、文本分类、语言模型等任务的基础。

本章首先以英文为例介绍词形分析任务以及主要的词性分析算法，然后以中文分词为例介绍词的切分问题和常见分词算法，最后介绍词性分析任务以及分析算法。

3.1 词形分析 . . . .	3
3.2 词语切分 . . .	11
3.3 词性标注 . . .	24
3.4 延伸阅读 . . .	29
3.5 习题 . . . . .	30

## 3.1 词形分析

词通常是由**语素（Morpheme）**构成。**语素**是一个语言中最小的音义结合体。在词汇系统中语素是最小的有意义的语言单位。例如：“电灯”，包含“电”和“灯”两个语素；在英文单词 dog 末尾添加 s 可以将它从单数名词变成复数名词 dogs；德语单词 bäcker 末尾添加 in 可以将它从阳性词（男面包师）变为阴性词 bäckerin（女面包师）。只包含一个语素的词语称为**简单词（Simple Word）**，而包含多个语素的词称为**复杂词（Complex Word）**。在语言学中，研究单词的

语素又称词素

内部结构及其构成方式的学科称为**形态学 (Morphology)**。将一个词分解成为语素的过程称为**词形分析 (Morphological Parsing)**。

形态学又称为构词学

一个词的语素可以主要分成两类: **词根 (Lemma)** 和**词缀 (Affix)**。词根也称为原形或字典形, 指能在字典中查到的语素, 通常是一个词最主要的语素。词缀是其他附着在原形上语素, 帮助在原形基础上衍生出新词, 包含前缀和后缀等。例如:

单词 cats 中, ‘‘cat’’ 为原形, ‘‘-s’’为后缀

单词 unhappy 中, ‘‘happy’’为原形, ‘‘un-’’为前缀

一个词也可以包含多个词缀, 例如: unhappiness 包含前缀 “un-” 和后缀 “-ness”。同样, 一个词也可以包含多个词根, 例如: homework 包含词根 “home” 和 “work”。有些语言的单词通常只包含一个或者两个词素, 但是有一些语言的单词则包含多达十个以上的词素。中文中每个单词的词素都很少, 也不会根据性、数、格、人称等发生形态变化。

本节将首先以英语为例介绍词形的变化形式以及词形分析任务, 然后介绍基于有限状态转换机的词形分析算法。

## 英语词形变化概述

英语中的很多词都包含两个或两个以上的语素, 英语的词形变化主要通过以下四种方式:

- **屈折变化 (Inflection)** 是指通过 “词根 + 词缀” 的方式构成和原形 “同一类型” 的词。同一类型指词义和词性没有发生明显的变化, 或者说通过屈折变化得到的词的词义与它的原形相似。例如:

在名词后加 -s 后缀构成复数名词 (cat+s)

在动词后加 -ed 后缀构成动词的过去式 (walk+ed)

► **派生变化 (Derivation)** 是指通过“词根 + 词缀”的方式构成和原形“不同类型”的词。例如：

employ 添加后缀 -ee 变为 employee  
employ 添加后缀 -er 变为 employer  
meaning 添加后缀 -less 变为 meaningless

可以看到，增加后缀后，词根的词义发生了较为明显的变化。此外，通过添加词缀的方式也可以使得词的词性发生变化。例如：

形容词可以组合 -ize 后缀变为动词 (medical/medicalize)  
名词可以组合 -al 等后缀变为形容词 (sensation/sensational)

► **组合 (Compounding)** 是指通过组合多个词根构成一个新词。例如：

homework 是由 home 和 work 组合而成  
waterproof 是由 water 和 proof 组合而成

► **附着 (Cliticization)** 是指“词根 + 附着语”的方式。附着语通常在语法上等同于一个词，通过特殊的方式“附着”在词根上。例如：

I'm 中的 'm 代表 am 附着在 I 上  
We're 中 're 代表 are 附着在 We 上

除此之外，英语中还有一些次要的形态变化，包括词语新造 (Coinage)、截搭 (Blending)、逆构 (Backformation)、缩略 (Acronym)、截短 (Clipping) 等。上述形态变化的定义和例子如表3.1所示。

更详细的词和形态学相关工作可以参考《语言学教程》第三章相关内容 [1]。

[1]: 胡壮麟 (2015), 语言学教程 (第五版中文本)

形态变化	描述	例子
词语新造 (Coinage)	完全新造的词语	iPhone
截搭 (Blending)	两个词语各自的一部分拼接	smoke + fog → smog
逆构 (Backformation)	简单词感知为多个语素组成	editor → edit
缩略 (Acronym)	短语中多个单词首字母拼接	United Nations → UN
截短 (Clipping)	长单词截短	demonstration → demo

表 3.1: 英语中其它词形变化。

### 词形分析算法

词形分析的一个简单的方法是将每一个词的所有词形变换都存储下来,使用时直接匹配查找。对于英语来说,一个包含所有词形的词典能够较为有效的支撑许多应用场景。但是由于用词方式变化和新词的不断出现,对这个字典需要进行及时维护。同时,对于一些语言(特别是土耳其语,阿拉伯语等黏着语)枚举所有词的词形变换则是不可能的。例如:土耳其语词汇 `uygarlatramadklarmzdanmsnzcasna`[2] 是由以下 10 项变换组合而成:

`uygar +la +tr +ama +dk +lar +mz +dan +m +snz +casna`  
`civilized +BEC +CAUS +NABL +PART +PL +P1PL +ABL +PAST +2PL +AsIf`

其中除了词根 `uygar` 以外,其他语素的含义如下:

- +BEC “变成”(become)
- +CAUS 标识使役动词
- +NABL “不能”(not able)
- +PART 过去分词
- +PL 名词复数
- +P1PL 第一人称复数所有格
- +ABL 表来源的离格\*(ablative (from/among) case maker)
- +PAST 带过去时的间接引语 (indirect/inferential past)
- +AsIf 从限定动词 (finite verb) 派生出的副词

[2]: Jurafsky et al. (2008), *Speech and Language Processing: An Introduction to speechrecognition, natural language processing and computational linguistics*



我们可以看到，由于词性变换的复杂性，一个词的原形可能衍生出很多不同的词。因此，设计更有效率的词形分析算法是十分必要的。

**Definition 3.1.1** 词形分析算法任务是对于给定的词语  $w$ ，输出构成该词的词根以及词形特征集合。

表3.2给出了典型词形分析算法的输入和输出结果样例。

输入	输出
cats	cat+N+pl
cat	cat+N+sg
buses	bus+N+pl
goose	goose+N+sg
geese	goose+N+pl
walk	walk+N+sg
walk	walk+V
walks	walk+V+3sg
walking	walk+V+prespart
walked	walk+V+past
walked	walk+V+pastpart
take	take+N+sg
took	take+V+past
taken	catch+V+pastpart

表 3.2: 词形分析输入输出示例

从表3.2可以看到，词形分析的结果除了包含词根外，还包含一些词形特征 (Morphological features)。例如：“+N”(名词)，“+V”(动词) 表示词性，“+pl”表示复数，“+sg”为单数，“+3sg”表示第三人称单数，“+prespart”表示用于进行时的 ing 形式，“+pastpart”表示用于完成时或被动语态的 -ed/-en 形式。值得注意的是，对于同一个词可以有不同的词形分析结果（例如“walk”）。这些不同的词形分析结果与词所在的上下文环境有关。这里主要讨论与上下文无关的分析算法。

## 基于有限状态转换机的词形分析

有限状态转换机 (Finite State Transducer, FST) 是有限状态机 (Finite State Automata, FSA) 的扩展。对于一个输入串, 有限状态机在每个输入字符后进行状态转移。与有限状态机不同, 有限状态转换机在状态转移的同时, 也会输出一个字符。换句话说, 有限状态机能够识别一个输入串, 有限状态转换机能够将输入串转换到一个输出串。一个有限状态转换机的定义包含以下 7 个部分:

- ▶  $\Sigma$ : 一个有限的字符集合 (输入字符集)
- ▶  $\Gamma$ : 一个有限的字符集合 (输出字符集)
- ▶  $Q$ : 状态集合
- ▶  $q_0 \in Q$ : 初始状态
- ▶  $F \subseteq Q$ : 最终状态集合
- ▶  $\delta(q, w)$ : 状态转移函数。对于给定的状态  $q \in Q$ , 输入串  $w \in \Sigma^*$ ,  $\delta(q, w) \subseteq Q$  表示在状态  $q$  下接收输入  $w$  后, 所有可能的下一个状态
- ▶  $\sigma(q, w)$ : 输出函数。对于给定的状态  $q \in Q$ , 输入串  $w \in \Sigma^*$ ,  $\sigma(q, w) \subseteq \Gamma^*$ , 表示在状态  $q$  下接收输入  $w$  后, 所有可能的输出字符串

图3.1为一个简单的有限状态转换机的示例, 其中  $q_2$  为终态。  $x : y$  表示当输入串是  $x$  时, 输出  $y$ 。  $x$  为  $x : x$  的简写。

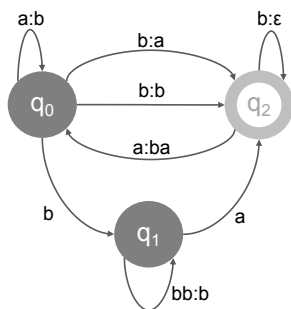


图 3.1: 有限状态转换机 FST 示例

这里定义的有限状态转换机是**非确定的** (non-deterministic), 其非确定性主要有三个来源:

- ▶ 状态转移函数  $\delta$  在同一个状态下接收同一个输入可以转移到不同的状态 (如图3.1中的  $q_0$  状态)。
- ▶ 输出函数  $\sigma$  在同一个状态下接收同一个输入可以输出不同的字符串
- ▶ 可以输入输出空串  $\varepsilon$ 。

与 FSA 不同, 不是任意一个非确定有限状态转换机都能够对应到一个确定有限状态转换机。这导致在使用有限状态转换机进行字符串转换的过程通常需要使用搜索算法。另一方面, 在实际应用中通常可以增加有限状态转换机的确定性。例如, **有序转换机** (Sequential FST) 限定状态转移函数  $\delta$  的输出是一个确定的状态且输入字符集  $\Sigma$  不包含空串  $\varepsilon$ ; **次有序转换机** (Subsequential FST) 在有序转换机的基础上, 在到达每一个终态后可以输出一个字符串; 以及  **$p$ -次有序转换机** ( $p$ -subsequential FST) 次有序转换机的终态可以输出  $p$  个字符串。

为了构造基于有限状态转换机词形分析器, 我们将会用到以下有限状态转换机操作:

- ▶ 取交集操作  $\cap$ : 有限状态转换机  $T_1$  和  $T_2$  的交  $T_1 \cap T_2$  接收  $T_1$  与  $T_2$  所接受的输入输出对的交集。
- ▶ 取逆操作  $T^{-1}$ : 将  $T$  的输入输出互换得到一个新的有限状态转换机。
- ▶ 复合操作  $\circ$ : 假设  $T_1$  将输入字符串集合  $I_1$  映射到输出字符串集合  $O_1$ ,  $T_2$  将输入字符串集合  $O_1$  映射到输出字符串集合  $O_2$ , 它们的复合  $T_1 \circ T_2$  将  $I_1$  映射到  $O_2$ 。

对于词形分析任务来说, 我们考虑它的逆问题 (称为**生成问题**): 给定原形和词形特征 (表3.2中第二列), 输出词 (表3.2中第一列), 通过构造生成问题的有限状态转换机, 可以使用取逆操作得到词形分析的有限状态转换机。下面将以名词的单数, 复数屈折变换为例, 简要介绍构造生成问题有限状态转换机的过程。

首先，根据规则名词单数，不规则名词单数，不规则名词复数构造 3 条生成路径，如图3.2中的转换机  $FST_{mid}^{feat}$ 。  $FST_{mid}^{feat}$  中“regular noun”，“irregular singular noun”和“irregular plural noun”可以进一步在词典中展开 (如图3.3)。

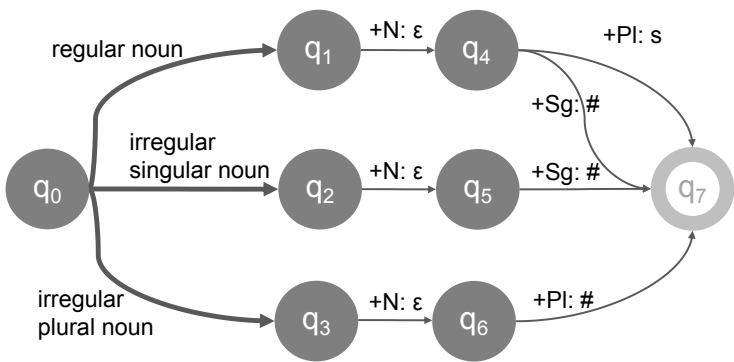


图 3.2:  $FST_{mid}^{feat}$  根据单数复数词形特征生成一种名词的中间表示形式。# 表示单词结尾, 语素间用 ^ 分割。

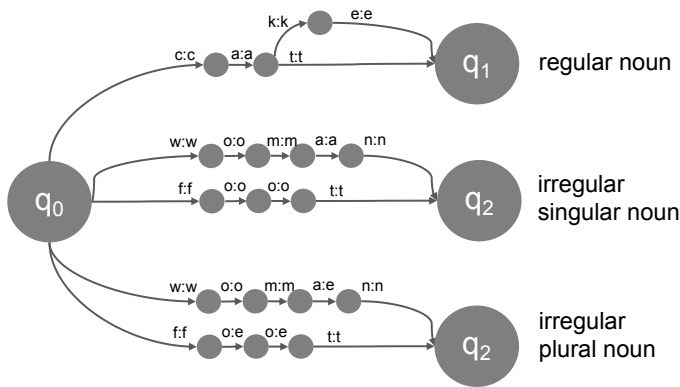


图 3.3: 图3.2中“regular noun”，“irregular singular noun”，“irregular plural noun”的展开。

通过图3.2和图3.3，可以将词形特征转化成为中间表示。例如：

- “cat+N+Pl” 转换成 “cat^s#”
- “foot+N+Sg” 转换成 “foot#”
- “foot+N+Pl” 转换成 “feet#”
- “box+N+Pl” 转换成 “box^s#”

注意到对于规则名词 “box”(以及其他以 “z, s, x” 结尾的规则名词)，在生成它的复数形式时，需要修改拼写方式：需要插入 “-es” 后缀而非 “-s” 后缀。

图3.4为一个有限状态转换机，实现以中间表示为输入，输入修改过的后缀，即将“box<sup>^</sup>s#”转换成“boxes#”，同时对其他词的拼写不做改动（“cat<sup>^</sup>s#”转换成“cats#”）。类似的拼写改动还包括以“y”结尾的词需要把中间表示中的“-s”后缀修改为“-ies”后缀。每一个规则  $r$  对应于一个  $FST_r$ 。

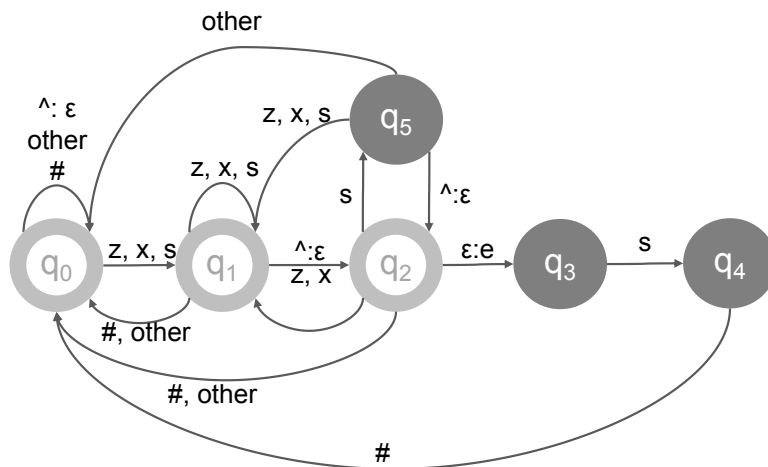


图 3.4: 使用有限状态转换机实现在“z, s, x”后插入 e。q<sub>1</sub> 表示观察到“z, s, x”。在输入语素分隔符“^”后，转移到状态 q<sub>2</sub>。q<sub>2</sub> 实现插入“e”，转移到 q<sub>3</sub>，同时，当 q<sub>2</sub> 输入“s”，后转移到 q<sub>3</sub>，如果此时输入单词解释符 #，则判断为非法（即不符合“在 z, s, x 后应插入 e”的拼写规则），并拒绝接收这样的字符串。

最后，可以通过有限状态转换机的操作将  $FST_{mid}^{feat}$  组合  $FST_r$  到生成问题的有限状态转换机：

$$cFST_{gen} = FST_{mid}^{feat} \circ (\cap_r FST_r)$$

同时，通过取逆操作可以得到词形解析的有限状态转换机：

$$cFST_{parse} = FST_{gen}^{-1}$$

## 3.2 词语切分

如前所述，以汉语为代表的汉藏语系与以英语为代表的印欧语系不同，词之间不存在明显的分隔符，而是由一串连续的字符构成的句子。但是，由于自

然语言处理中通常以词为最小的处理单位，因此，对于汉语等语言需要首先进行分词。分词的效果也将直接影响词性、句法分析等任务的分析效果。

本节将以汉语为例介绍词语切分的基本概念以及所面临的主要问题，然后介绍基于词典、基于字统计、基于词统计以及基于神经网络的分词算法，最后介绍常见的中文分词数据集合。

## 中文分词概述

汉语作为汉藏语系的典型代表，其句子并不使用分割符来标识文本中的词。例如，本节标题“中文分词概述”一句中的六个汉字对应到三个汉语词汇（“中文”，“分词”，“概述”）。如何将汉字序列中的词切分出来是中文分词任务的核心目标。中文分词的主要困难来自以下三个方面：分词规范、歧义切分和未登录词识别。

汉语中对词的具体界定是一个目前还没有定论的问题。1992 年国家标准局颁布的《信息处理用现代汉语分词规范》中大部分规定都是通过举例和定性描述来体现。例如：“二字或三字词，以及结合紧密、使用稳定的二字或三字词组，一律为分词单位。”然而在实际应用中对“紧密”与“稳定”都很难界定。语言固有的歧义性是影响自然语言形式化建模的一大障碍。

对中文分词任务，汉字序列的歧义使同一个中文句子可以呈现出不同的分词结果。这些不同的分词结果也被称为切分歧义。例如：“南京市长江大桥”的正确词切分方式为“南京市 | 长江大桥”，但是也可能被切分为“南京 | 市长 | 江 | 大桥”。

未登录词 (*Out Of Vocabulary*, OOV) 是另外一个影响中文分词性能的重要因素。未登录词是在词典或者训练语料中没有出现的词。由于中文分词器并不识得这些词，所以在使用过程中往往会带来切分错误。事实上，相比于歧义切分问题，在真实应用环境中由未登录词引起的分词错误比例更高。

## 歧义切分

通常汉语中常见的切分歧义可以归纳为三类：交集型切分歧义、组合型切分歧义和真歧义。

**交集型切分歧义** 汉字串  $AJB$  称作交集型切分歧义，如果满足  $AJ$ ,  $JB$  同时为词 ( $A$ ,  $J$ ,  $B$  分别为汉字串)。此时汉字串  $J$  称作交集串。

例如：乒乓球拍卖完了。该例句中存在交集型切分歧义， $A$ ,  $J$ ,  $B$  分别代表“球”，“拍”和“卖”。“球拍”和“拍卖”同时都为合法词汇，它们之间存在有一个交集串。类似的例子还包括：“今天下雨”，“很多云彩”，“北京城市规划”，“中国产品质量”等。

**组合型切分歧义** 汉字串  $AB$  称作多义组合型切分歧义，如果满足  $A$ ,  $B$ ,  $AB$  同时为词。

例如：他马上过来。该例句中“马上”为组合型切分歧义。 $A$ ,  $B$ ,  $AB$  分别代表“马”，“上”和“马上”。类似的情况还包括：“才能”，“应对”，“学会”等。

**真歧义** 汉字串  $ABC$  称作真歧义，如果满足多种切分方式下语法和语义均没有问题，只有通过上下文环境才能给出正确的切分结果。

例如：白天鹅在水里游泳。对这个句子来说，两种不同的分词结果分别为“白天 | 鹅 | 在 | 水 | 里 | 游泳”以及“白天鹅 | 在 | 水 | 里 | 游泳”。这两种切分方式在语法和语义上都是正确的，需要考虑上下文环境才能进行正确判断。

交集型切分歧义也被称为偶发歧义，原因在于当两个有交集的词“偶然”的相邻出现时这样的歧义发生。而组合性切分歧义也称为固有歧义，原因在于组合歧义的是词固有的属性，不依赖于“偶然”发生的上下文。另外，这两种歧义切分的定义都是从机器识别的角度出发的。而事实上，许多歧义切分通常不会或者很少出现在真实中文文本中。例如，“平淡”根据定义属于组合型切分歧义，但实际上“平 | 淡”这样的切分方式在真实的上下文环境中非常罕见。根据 [3]

中的统计，中文文本中每 100 个词约出现 1.2 次切分歧义，其中交集型切分歧义和组合型切分歧义的比例约为 12:1。

[3]: 梁南元 (1987), “书面汉语自动分词系统—CDWS”

未登录词

汉语文本中常见的未登录词可以分为以下四类：

- ▶ 新出现的普通词汇。语言的使用会随着时代的变化而演化出新的词，这个过程在互联网环境中显得更为快速。例如: 下载, 给力, 点赞, 人艰不拆等。
- ▶ 命名实体 (Named Entity)。这类词具体包括 ①人名 (例如: 奥巴马, 周杰伦); ②地名 (例如: 外高桥, 张江); ③组织机构名 (例如: 亚洲善待博士组织, 中华人民共和国审计署驻上海特派员办事处); ④时间和数字 (例如: 2018-11-13, 正月初四, 110 亿人民币);
- ▶ 专业名词。这类词只出现在专业领域的新词 (例如: 图灵机, 碳 9, 埃博拉);
- ▶ 其他专有名词。例如: 新出现的产品名、电影名、书籍名等。

在新闻语料上的统计实验发现，几乎所有的分词错误都与未登录词相关。而其中由命名实体引起的分词错误占到了 50% 左右 [4]。因此，未登录词是中文分词的一个主要性能瓶颈。

[4]: 宗成庆 (2013), 统计自然语言处理

中文分词方法

中文分词任务可以定义为：给定一个中文句子  $x = c_1, c_2, \cdots, c_n$ , 其中  $c_i, 1 \leq i \leq n$  为字 (如表 3.3所示), 输出是一个词序列  $y = h(x) = w_1, w_2, \cdots, w_m$ , 其中  $w_j$  是一个中文词 (如表3.3所示)。

今	晚	的	长	安	街	流	光	溢	彩	。
$c_1$	$c_2$	$c_3$	$c_5$	$c_5$	$c_6$	$c_7$	$c_8$	$c_9$	$c_{10}$	$c_{11}$
今晚		的	长安街		流光溢彩		。			
$w_1$		$w_2$	$w_3$		$w_4$		$w_5$			

表 3.3: 中文例句及其分词结果。



中文分词方法可以分为无监督分词方法和有监督分词方法两大类。无监督分词方法通常需要依赖词典信息，而有监督分词方法则将分词转换为有监督分类问题，利用已标注中文分词结果的语料上构造统计模型。在本节中我们将对上述方法分别进行介绍。

基于词典的分词方法

基于词典的无监督分词主要包含前向最大匹配，后向最大匹配以及双向最大匹配等三大类。给定一个中文句子，这些算法试图从词典中找到针对该句子最好的分词方案。

前向最大匹配算法的基本思想是，从左向右扫描句子，对于句子中的一个位置  $i$ ，依次考虑子串  $c[i : i + L - 1], c[i : i + L - 2], \dots, c[i : i]$ ，其中  $c[i : j] \triangleq c_i c_{i+1} \dots c_j$  表示从第  $i$  个字到第  $j$  个字构成的字串 (每一个这样的字串对应于一个候选的词)， $L$  表示词典中词的最大长度。当某一个  $c[i : j]$  能够对应字典中的一个词时，输出这个词并从  $j + 1$  开始重复以上的过程直至整个句子被遍历完成。

例如，对表3.3中的句子，前向最大分词的过程如下 (为简单起见，假设词典中的词表为 {“今晚”，“晚上”，“的”，“长安街”，“长安”，“流光溢彩”，“。”})

时间步	句子位置	候选匹配	选择词
1	1	今晚的长, 今晚的, 今晚, 金	今晚
2	3	的长安街, 的长安, 的长, 的	的
3	4	长安街流, 长安街, 长安, 长	长安街
4	7	流光溢彩, 流光溢, 流光, 流	流光溢彩
5	11	。 eos eos eos, 。 eos eos, 。 eos, 。	。

表 3.4: 表3.3中句子的前向最大匹配分词过程。eos 表示句子结尾。

后向最大匹配和正向最大匹配思想相同，区别在于对于句子从右向左扫描。双向最大匹配则是同时进行前向最大匹配和反向最大匹配，当两者的分词结果不同时，可以使用启发式的规则决定选取哪一个作为最终的输出（例如选择平均词长较大的一个）。

可以看出，基于词典的分词方法具有简单，快速，可控等优点。但对于没有在词典中出现的词没有很好的处理方案。基于统计的模型能够在一定程度上弥补它们的缺陷。

基于字的统计模型

基于统计的中文分词算法通常使用序列标注模型建模。具体来说，对希望输出的词序列使用一个标签序列编码。对每一个字  $c_i$ ，根据它在分词结果中的位置，可以赋予标签  $y_i$ 。例如“BIES”标签系统，

- ▶  $y_i = B$  表示  $c_i$  处于一个词的开始位置。
- ▶  $y_i = I$  表示  $c_i$  处于一个词的中间位置。
- ▶  $y_i = E$  表示  $c_i$  处于一个词的结束位置。
- ▶  $y_i = S$  表示  $c_i$  单独构成一个词。

表3.3的所对应的序列编码如表3.5所示。经过序列编码后，输出目标  $y$  也变为长度为  $n$  的“BIES”标签序列  $y = w_1, w_2, \cdots, w_m = y_1, y_2, \cdots, y_n$ 。

今晚	的	长安街	流光溢彩	。
BE	S	BIE	BIIE	S

表 3.5: 使用“BIES”标签对词序列编码。

序列标注问题可以采用条件随机场 (Conditional Random Field (CRF)) 等结构化机器学习方法进行解决。在条件随机场模型中，通常假设特征函数  $\varphi(x, y_i, y_{i-1})$  都仅依赖于输入  $x$  和相邻的两个标签  $y_i, y_{i-1}$ (也称为一阶马尔可夫假设)。这样的假设虽然牺牲了一定的特征表示能力，但是同时也使得序列标注模型的训练和解码能够较为高效的完成。

条件随机场模型的详细介绍详见本书第XXX章。

如何设计有效的  $\varphi(x, y_i, y_{i-1})$  对于序列标注任务是至关重要的。针对中文分词问题，我们介绍一种基于模板的稀疏特征表示方法。在基于模板的特征表

示中，特征函数  $\varphi(x, y_i, y_{i-1})$  的每一维为一个 0,1 取值的函数。例如，在中文分词任务中一个典型的特征如下：

$$\varphi_k(x, y_i, y_{i-1}) = \begin{cases} 1 & \text{if } x_i = c \text{ and } y_i = \text{"B"} \text{ and } y_{i-1} = \text{"E"} \\ 0 & \text{otherwise} \end{cases}$$

其中， $c$  为一个中文字。这里针对所有可能的中文字  $c$  (或者训练集中出现的字) 都有一个对应的维度 (即，这个特征模板将展开为长度为字典长度的独热向量 (one-hot vector))。表3.6列出了中文分词任务常用的模板。

模板名	描述	例子
$x_i$	当前字	安
$x_{i-1}$	$i - 1$ 位置的字	长
$x_{i-2}$	$i - 2$ 位置的字	的
$x_{i+1}$	$i + 1$ 位置的字	街
$x_{i+2}$	$i + 2$ 位置的字	流
$x_{i-2}, x_{i-1}$	$i - 2$ 开始的 bigram	长安
$x_{i-1}, x_i$	$i - 1$ 开始的 bigram	的长
$x_i, x_{i+1}$	$i$ 开始的 bigram	安街
$x_{i+1}, x_{i+2}$	$i + 1$ 开始的 bigram	街流

表 3.6: 中文分词常见模板。“例子”一栏中包含对应模板在表3.3中句子的第 5 个位置时的取值。

基于词的统计模型

基于词的中文分词任务定义为寻找一个将输入句子  $x \in X$  转换为单词序列  $y \in Y$  的映射，其中  $X$  是可能的原始输入问句集合， $Y$  是可能的输出问句集合，该映射用  $F(x)$  表示，公式可表达为：

$$F(x) = \arg \max_{y \in GEN(x)} Score(y)$$

其中  $GEN(x)$  代表对于每一个输入句子  $x$  可能的所有候选输出。

打分函数  $Score(y)$ ，针对每一个分词后的句子  $y$  定义一个全局特征向量  $\Phi(y) \in \mathbb{R}^d$ ，其中  $d$  代表模型中的特征数量。函数  $Score(y)$  由矢量  $\Phi(y)$  和一组

参数  $\bar{\alpha} \in \mathbb{R}^d$  间的点积构成,  $a_i$  代表第  $i$  个特征的参数:

$$Score(y) = \Phi(y) \cdot \bar{\alpha}$$

对于参数  $\bar{\alpha}$ , 可以使用感知机算法进行训练。对每一句句子进行解码得到一组候选分词结果的集合, 对于集合中的每一句经过分词的句子, 将之与正确答案进行比对, 如果结果错误则更新参数  $\bar{\alpha}$ 。

---

**Algorithm 1:** 基于感知机算法的 Score 函数训练

---

```

1: 输入: 训练数据  $(x_i, y_i)$ 
2: 输出:  $\bar{\alpha}$ 
3: 初始化参数  $\bar{\alpha} = 0$ 
4: for  $t = 1$  to  $T$  do
5:   for  $i = 1$  to  $N$  do
6:      $z_i = \arg \max_{y \in GEN(x_i)} \Phi(y) \cdot \bar{\alpha}$ 
7:     if  $z_i \neq y_i$  then
8:        $\bar{\alpha} = \bar{\alpha} + \Phi(y_i) - \Phi(z_i)$ 
9:     end if
10:  end for
11: end for
12: return  $\bar{\alpha}$ 

```

---

在进行解码的过程中, 每一个句子都有指数级数量的候选分词结果, 如果将所有可能的结果都枚举一遍的话, 搜索空间将变得非常巨大, 使得我们无法有效地进行训练与推断。

对于这一问题, 常见的解决方式是使用 Beam Search 算法进行解码。Beam search 是一种常用的限制搜索空间的启发式算法, 在每一步解码过程中, 从上一步解码的所有候选结果中选取前  $K$  个得分最高的结果继续解码, 而舍弃得分排在第  $K$  名之后的所有候选结果。Beam search 可以理解作为一种“松弛”过的贪心算法, 它并不能保证得到一定会得到得分最高的候选解码序列, 但往往可以得到想要的答案。算法2给出了应用于中文分词的 Beam Search 算法详细流程。

用一个例子来理解分词中的 Beam Search 算法: 假设有这样一句话“奥巴

---

**Algorithm 2:** Beam Search 解码

---

```

1: 输入: 原始句子  $s$ 
2: 输出:  $src$ 
3: 初始化  $src = [[]], tgt = []$ 
4: for  $index = 0$  to  $s.length - 1$  do
5:    $var char = s[index]$ 
6:   for  $item$  in  $src$  do
7:      $var item_1 = item$ 
8:      $item_1.append(char.toWord())$ 
9:      $tgt.insert(item_1)$ 
10:    if  $item.length > 1$  then
11:       $var item_2 = item$ 
12:       $item_2[item_2.length - 1].append(char)$ 
13:       $tgt.insert(item_2)$ 
14:    end if
15:  end for
16:   $src = tgt$ 
17:   $tgt = []$ 
18: end for

```

---

马访问北京”，Beam 大小为 2，在解码到第 6 个字之前的候选集中将会有两个候选分词结果:

“奥巴马/访问”  
“奥巴/马/访问”

对于第六字“北”可以扩展出 4 个新的候选分词句:

“奥巴/马/访问/北”  
“奥巴/马/访问北”  
“奥巴马/访问/北”  
“奥巴马/访问北”

经过打分排序后的结果为:

“奥巴马/访问/北”  
“奥巴/马/访问/北”  
“奥巴马/访问北”  
“奥巴/马/访问北”

由于 Beam 大小设置为 2，因此只取头两句子继续解码，舍弃之后的句子。余下步骤依此类推，从而得到最终的结果。可以注意到，在每一步只做了 4 次解码操作，从而极大地降低了计算开销。

输入特征由一系列人工选取的特征值组成，包含字、词以及长度信息。在训练时会使用特征模板将解码得到的序列映射到特征向量，特征向量将被输入到评分函数中。具体的特征模板如下表所示：

1	单词 $w$
2	二元单词 $w_1w_2$
3	单字符单词 $w$
4	初始字符 $c$ 以及长度 $l$
5	终止字符 $c$ 以及长度 $l$
6	由空格隔开的字符 $c_1$ 和 $c_2$
7	二元字符 $c_1c_2$
8	所有单词的第一个与最后一个字符 $c_1$ 和 $c_2$
9	字符 $c$ 的前一个词 $w$
10	单词 $w$ 之后的第一个字 $c$
11	两个连续单词的第一个字符 $c_1$ 和 $c_2$
12	两个连续单词的最后一个字符 $c_1$ 和 $c_2$
13	单词长度 $l$ 以及之前的词 $w$
14	单词的长度 $l$ 以及之后的单词 $w$

表 3.7: 输入特征模板

基于神经网络的分词

随着深度学习技术的发展，许多中文分词算法也采用了基于神经网络模型。图3.5给出了一个使用 BiLSTM+CRF 进行分词的框架。BiLSTM+CRF 是融

合 LSTM 和 CRF 一种常用于序列标注任务的框架，可以有效的结合结构化学习和神经网络的特点，在很多自然语言处理任务上都取得了很好的效果。

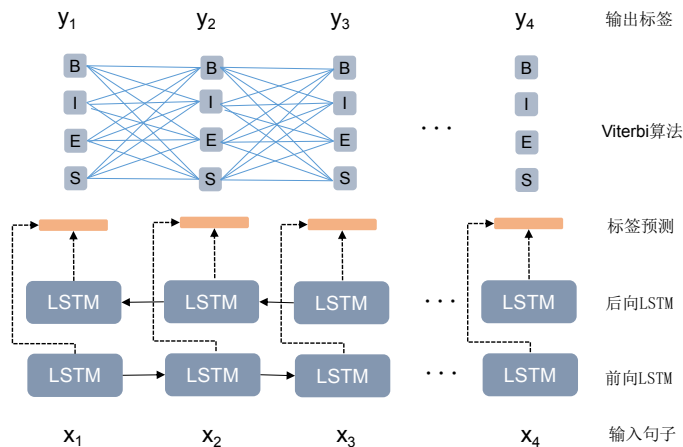


图 3.5: 基于 BiLSTM+CRF 的神经网络分词模型。

如图3.5所示，BiLSTM-CRF 主要包含三层：输入层、BiLSTM 层和 CRF 层。在中文分词任务中，我们可以采用与基于字的统计方法（参见3.2）类似的问题建模方法，将分词任务转换为字的序列标注任务，即对每一个字  $c_i$ ，根据它在分词结果中的位置以及所采用标签系统（例如：“BIES”等），赋予标签  $y_i$ 。在输入层，需要将每个字转换为低维稠密的字向量（character embedding） $x_i$ 。第二层 BiLSTM 层采用双向 LSTM，其主要作用是提取句子特征。将句子中的每个字向量序列  $(x_1, x_2, \dots, x_n)$  输入到双向 LSTM 各个时间步，再将正向 LSTM 输出的隐状态序列  $(h_1, h_2, \dots, h_n)$  与反向 LSTM 隐状态序列的  $(h_1, h_2, \dots, h_n)$  按位置进行拼接  $h_i = [h_i; h_i]$ ，从而得到完整的隐状态序列。在此基础上，我们可以通过第四章所介绍 Softmax 函数，得到每个字  $c_i$  独立进行进行的分类标签预测情况。之后结合第三层 CRF 层，对整个句子序列的输出进行建模和预测。

## 中文分词语料

如前文所述，现代分词系统通常常常需要依赖大规模标注语料。本节将介绍当前广泛使用的中文分词语料库。

### 北京大学分词语料库（PKU）

该数据集是由北京大学计算语言学研究所与富士通公（Fujitsu）合作在 110 万字《人民日报》原始数据基础上，进行了分词的信息，字符总数量约为 182 万。

示例：在 / 1998 年 / 来临 / 之际，我 / 十分 / 高兴 / 地 / 通过 / 中央 / 人民 / 广播 / 电台 / 、中国 / 国际 / 广播 / 电台 / 和 / 中央 / 电视台，向 / 全国 / 各族 / 人民，向 / 香港 / 特别 / 行政区 / 同胞、澳门 / 和 / 台湾 / 同胞、海外 / 侨胞，向 / 世界 / 各国 / 的 / 朋友 / 们，致以 / 诚挚 / 的 / 问候 / 和 / 良好 / 的 / 祝愿！

同时他们还制定了《现代汉语语料库加工规范》，在该规范中，规定了分词要与词性标注进行结合的原则。例如，“复合”方式可将两个构词成分结合成一个新词。构词成分通常认为是语素。由于复合词的构成方式和短语的构成方式是一样的，包括定中、状中、述宾、述补、主谓、联合、连动等。当语素是成词语素时，复合词与短语的界限是不清晰的。只有当构词成分中至少有一个是不成词语素时，才有把握判断新组合的结构是一个未登录词，否则存在一定的弹性。形式上，两个字的或三个字的组合可以较宽地认为是一个词。规范中规定了许多新词的构词方式，也规定了一般性名词和专有名词切分的规范

下载地址：<http://sighan.cs.uchicago.edu/bakeoff2005/>



### 香港城市大学分词语料库 (CITYU)

该数据集是香港城市大学语言资讯科学研究中心制作的繁体中文分词数据集，对包含 145 万字的原始数据进行了切分。

示例：一 / 宗 / 平常 / 的 / 超速 / 上訴 / ， 揭露 / 了 / 青嶼 / 幹線 / 一 / 隧 / 三 / 橋 / 的 / 80 / 公里 / 車速 / 上限 / 原來 / 並 / 沒有 / 刊憲 / ， / 立即 / 有 / 司機 / 組織 / 表示 / 考慮 / 提出 / 集體 / 訴訟 / ， 希望 / 取回 / 過往 / 因 / 超速 / 失去 / 的 / 分數 / 及 / 罰款 / ； 另一邊廂 / ， 警方 / 表示 / 會 / 考慮 / 上訴 / ， 並 / 堅稱 / 運輸署長 / 有權 / 在 / 毋須 / 刊憲 / 的 / 情況 / 下 / ， 在 / 青馬 / 管制區 / 實施 / 「 / 暫時 / 的 / 速度 / 限制 / 」 / 。

他们制定了相关的切词规则，在名词，数词，时间词，略语，二字结构，三字复合词，四字词，短语，叠词，非汉字部分这十个方面的切分进行了详细的规范。另外还对其他方面进行了补充，古语方言和熟语等不进行切分，例如踏破铁鞋无觅处这句话不进行分词。

下载地址：<http://sighan.cs.uchicago.edu/bakeoff2005/>

### 微软研究院分词语料库 (MSR)

该语料库是由微软亚洲研究院 (MSRA) 整理，在 230 万字的简体中文原始语料上进行划分，采用 CP936 的编码方式。

示例：产油国 / 、 / 国际 / 石油 / 公司 / 和 / 石油 / 消费 / 国 / 应该 / 相互 / 协商，在 / 长期 / 互利 / 基础 / 上 / 建立 / 新 / 的 / 油 / 价 / 体系。

数据集将词汇分为三大类，词汇词（如教授，高兴，吃饭），命名实体（如蒙特利尔，中央民族乐团）和陈述词。其中陈述词类别较多，有日期，时间，持续时间，量词电话号码等。

下载地址：<http://sighan.cs.uchicago.edu/bakeoff2005/>

语料库名称	数据集规模	语言	标注内容
PKU	110 万	简体中文	分词、词性、专有名词
CITYU	145 万	繁体中文	分词
MSR	230 万	简体中文	分词

表 3.8: 中文分词语料库汇总

### 3.3 词性标注

#### 词性标注概述

词性是根据词在句子中扮演的语法角色，以及和周围词的关系对词的一个分类，因此，词性也被称为词类 (part of speech, POS)。例如，通常表示事物的名字 (“钢琴”)，地点 (“上海”) 被归为名词，而表示动作 (“踢”)，状态 (“存在”) 的词被归为动词。通过词性，可以大致圈定一个词在上下文环境中有可能搭配词的范围 (介词 “in” 后面通常跟名词短语而非动词短语)，从而为语法分析，语义理解提供帮助。由此，词性也被称为带有 “分布式语法” 信息 (syntactic distributional properties)。

根据词性，词通常被划分成为两类: 开放词类 (open class) 和 封闭词类 (close class)，也被称为实词 (content words) 和虚词 (function words)。顾名思义，随着语言的应用，开放词类通常接纳新的词。例如，名词作为开放词类，名词的集合通常会不断的变化。比如，“区块链” 作为一个新技术的名字，被加入名词集合中。相比而言，封闭词类相对固定。例如，英文中介词集合 (“in”，“on” 等) 通常不随语言使用而产生大的变化。换句话说，想象两个不同领域的专家 (数学家和语言学家)，他们的论著中可能使用的开放词类差别交大，但对封闭词类的使用都是相同的。

在英文中，开放词类通常为名词，动词，形容词和副词。

- 名词 (Noun) 通常充当表示具体或者抽象事物的语法角色。包括大多数人，地点等。名词可以进一步分为专有名词 (Proper Noun) 和普通名词 (Common Noun)。专有名词通常表示一些特定的名称，例如特定公司名

“Facebook”，特定的工具名 “StarCraft” 等，相比而言 “book”，“table” 等为普通名词。名词也可以分为可数名词和不可数名词。可数名词通常有单数和复数两种词形变化，并且与它们相关联的冠词，动词也会不同。

- ▶ 动词 (Verb) 通常表示动作和状态。
- ▶ 形容词 (Adjective) 通常用于修饰名词或者名词短语，如 “beautiful”，“fast” 等。
- ▶ 副词 (Adverb) 是一类较为混杂的词类。通常情况下用来修饰动词，如 “carefully”，“strongly”。更进一步细分可以分为表方位的副词 (Locative adverb，表示一个方位以及和这个方位间的联系)。例如 “home”，“here”，“there” 等。表时间的副词 (Time adverb)，如 “already”，“soon”，“yesterday” 等。表程度的副词 (Degree adverb)，如 “quite”，“too”，“extremely” “perfectly” 等。表情态的副词 (Manner adverb，表示一个动作或过程的执行状态)，如 “slowly”，“powerfully”，“well”，“hard” 等。

英文的封闭词类的几个主要的类别如下，

- ▶ 介词 (Preposition)，如 “on”，“in”，“with” 等。
- ▶ 限定词 (Determiner)，包括冠词 (article，“a”，“an”，“the”)，指示代词 (Demonstrative，“this”，“that” 等)，物主代词 (Possessive Determiners，“my”，“your” 等)，数词 (Numeral，“one”，“two” 等)，量词 (Quantifiers，“all”，“some”，“many” 等)，相互代词 (Distributive pronoun，“each”，“other”，“any” 等)，疑问代词 (Interrogative determiners，“who”，“when” 等)。
- ▶ 连词 (Conjunction)，如 “and”，“but”，“if” 等
- ▶ 助词 (Auxiliaries)，如 “can”，“must”，“do” 等

表3.9列出了在宾州树库 (PTB) 中标注的词性。

表 3.9: 宾州树库中的词性标签

标签	描述	标签	描述
CC	并列连词	CD	数字
DT	限定词	EX	<i>there</i>
FW	外来词	IN	介词或从属连词
JJ	形容词	JJR	形容词比较级
JJS	形容词最高级	LS	列表项标记
MD	情态助动词	NN	名词单数
NNS	名词复数	NNP	专有名次单数
NNPS	专有名词复数	PDT	前限定词
POS	所有格结束词	PRP	人称代名词
PRP\$	物主代词	RB	副词
RBR	副词比较级	RBS	副词最高级
RP	小品词	SYM	符号
TO	<i>to</i>	UH	叹词
VB	动词	VBD	动词过去式
VBG	动词现在进行式	VBN	动词过去分词
VBP	动词一般现在式 非第三人称单数	VBZ	动词一般现在式 第三人称单数
WDT	Wh-限定词	WP	Wh-代词
WP\$	所有格 Wh-代词	WRB	Wh-副词

https:

[//www.ling.upenn.edu/courses/Fall\\_2003/ling001/penn\\_treebank\\_pos.html](http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html)

### 词性标注方法

词性标注任务是指给定一个句子  $(w_1, w_2, \dots, w_n)$ ，输出句子中每个词的词性。例如对于句子: I like reading books. 其对应的输出为: I/PRP like/VBP reading/VBG books/NNS ./。其中，斜线后的部分标识了词性 (使用 PTB 标准, 表3.9)。

词性标注的主要难点在于歧义性，即一个词可能在不同的上下文中有不同的词性。例如 “book” 可以表示名词 “书”，也可以表示动词 “预定”，“good” 可以表示形容词 “好”，也可以表示名词 “货物”，“China” 可以表示专有名词 “中国”，也可以表示普通名词 “瓷器” 等等。因此需要结合上下文来确定词在句子中表现出来的词性。另一方面，在 Brown 语料库上的统计发现，超过 80% 的词通常只有一个词性。也就是说，在完全忽略上下文的情况下，使用固定的一词性能达到 80% 的准确性。

由于词性表以及词性定义有许多不同的变种，词性标注的结果与这些标注密切相关。本节将主要以 PTB 标准为例。

为了更进一步的提升词性分类的性能，也可以构建统计模型。显然，词性标注问题也是一类序列标注问题，因此条件随机场、RNN、LSTM 以及融合预训练模型等方法能够直接的应用在词性标注上。

## 主要数据集

### 宾州大学树库, PTB

对于英语而言，宾州大学句法树库 (Penn TreeBank, PTB) 是最早形成一定规模的句法树库，它是一个短语结构句法树库，取自于标准新闻题材，总计五万规模的句子，为每个句子标注了词性以及短语结构句法树。WSJ-PTB 是 PTB 项目的一部分，是目前新闻语料上最常用的词性标注数据集。WSJ-PTB 原始数据来自于 1989 年的华尔街日报，按照 PTB(V2) 的标注策略进行标注，拥有一百多万个标注单词，48 种不同的词性标签。

示例:France/NNP's/POS unemployment/NN rate/NN was/VBD steady/JJ at/IN a/DT seasonally/RB adjusted/VBN 9.5/CD %/NN in/IN September/NNP ./, the/DT Social/NNP Affairs/NNPS Ministry/NNP said/ VBD ./.

一般来说，一个句子虽然表面上呈现词语的线性排列，其内部的成分组织是存在一定层次结构的。PTB 使用树这种数据结构来表示句子的层次结构，构建一个大型是树库，包含丰富的语言结构信息。经过处理后，除了 WSJ-PTB 之外，PTB 还发布了标注的 Brown 语料库。

下载地址: <https://catalog.ldc.upenn.edu/LDC99T42>

### 中文宾州树库, CTB

宾州大学汉书树库 (Chinese Penn Treebank) 是建立一个大型的中文句法标注语料库。该数据集基于短语结构，进行了短语结构、短语功能、空元素等

的标注。发展至今共 8.0 版，第一版的语料主要来自于新华社的文章，在第二版中加入了香港和台湾的语料，以保证语料的多样性。2005 年 1 月发布的 5.0 版本包含 507222 个词，824983 个汉字，以及 18782 个句子，是目前最常用的 POS 任务数据集。

示例：上海\_NR 浦东\_NR 近年\_NT 来\_LC 颁布\_VV 实行\_VV 了\_AS 涉及\_VV 经济\_NN、\_PU 贸易建设\_NN、\_PU 规划\_NN、\_PU 科技\_NN、\_PU 文教\_NN 等\_ETC 领域\_NN 的\_DEC 七十一\_CD 件\_M 法规性\_NN 文件\_NN。\_PU 确保\_VV 了\_AS 浦东\_NR 开发\_NN 的\_DEG 有序\_JJ 进行\_NN。\_PU

在 CTB 中，汉语词性被划分为 33 类，包括 4 类动词和谓语形容词，3 类名词，1 类处所词，一类代词，3 类限定词和数次，一类量词，1 类副词，1 类介词，8 类语气词和 8 类其他词。

下载地址：<https://catalog.ldc.upenn.edu/LDC2005T01>

### Universal Dependencies (UD)

UD 是一个为多种语言开发的跨语言一致的树库项目，标注了语言的词性信息，形态特征和依存关系，其目标是促进多语言解析器的开发、跨语言学习和从语言类型学的角度进行解析研究。UD 是一个开放协作的项目，目前共有超过 200 个贡献者提供了 70 多种语言上的 100 多个树库。

示例：sentence: The oboist Heinz Holliger has taken a hard line about the problems. original: DT NN NNP NNP VBZ VBN DT JJ NN IN DT NNS. universal: DET NOUN NOUN NOUN VERB VERB DET ADJ NOUN ADP DET NOUN.

对各种树库下的标记集的高级分析表明，大多数标记集都是非常细粒度的，并且是特定于语言的。UD 使用 Petrov 等人在 2011 年提出的一个跨语言统一的词性标注系统 petrov2011universal。他们提出了一个由十二个通用词类构成的标记集，包括 NOUN (名词), VERB (动词), ADJ (形容词), ADV (副词), PRON (专

有名词), DET (限定词和冠词), ADP (介词和后置词), NUM (数字), CONJ (连接词), PRT (小品词), ‘.’ (名词所有格) 和 X (其他)。这 12 个类涵盖了大多数语言中最常见的词性。除了标记集之外, 他们还为来自 22 个语言的 25 个不同的树库开发了一个从细粒度词性标记到这个通用标记集的映射。

下载地址: <https://universaldependencies.org/>

语料库名称	数据集规模	语言	标注内容
WSJ-PTB	117 万	英文	分词、词性、句法树
CTB	50 万	中文	分词、词性、句法树
UD	70 种语言	多语言	分词、词性、句法树

表 3.10: 词性标注语料库汇总

### 3.4 延伸阅读

关于中文分词, 我们在前面介绍了基于循环神经网络的方法, 循环神经网络能很好地利用字符级别特征建模上下文信息实现分词任务。实际上, 神经网络有着这非常灵活的结构化建模能力。想要进一步提升分词的性能, 通过设计网络结构有效地引入词语级别的特征非常重要。其中基于转移的模型用于分词能够有效地结合词语特征 [5], 并将传统的特征模版和神经网络自动提取的特征结合起来, 在神经网络自动提取的特征和传统的离散特征的融合方法做了尝试。结果表明, 通过组合这两种特征, 分词精度可以得到进一步提升。另一种引入词语特征的方法是栅格化循环神经网络 [6], 这种方法能将句子里的字与所有可能匹配的词语同时进行建模, 从而提升分词准确率。

在词性标注任务中基于循环神经网络的方法已经能取得非常好的效果。如何提升词性标注的效率便成了研究者关注的问题。比如基于空洞卷积 [7]) 的词性标注利用卷积神经网络并行性能, 有能用空洞卷积的形式扩大感受野, 在取得较好准确率的同时也能有更快的处理速度。也有研究者将循环神经网络设计为并行结构——并行隐状态循环神经网络 [8], 同时引入全局结点来弥补上下文

[5]: Zhang et al. (2016), “Transition-Based Neural Word Segmentation”  
[6]: Yang et al. (2019), “Subword Encoding in Lattice LSTM for Chinese Word Segmentation” et al. (2017), “Fast and Accurate Entity Recognition with Iterated Dilated Convolutions”  
[8]: Zhang et al. (2018), “Sentence-State LSTM for Text Representation”

建模的不足，这种方法能打破了循环神经网络序列建模句子的方式，实现并行快速处理。

### 3.5 习题

1. 实现一个双向最大匹配分词方法。当前向和后向分词的结果不同时尝试设计一些消歧策略。
2. 中文分词中歧义切分包含几种主要的类别？针对每种歧义类别请试举几例，并说明具有歧义的切分方式。
3. 在中文分词中我们也可以使用 BIO 标签来建模序列标注 (标识一个词的开始, 中间和结束). 请尝试分析使用这样的标签集合与 BIES 标签集合会有什么区别? 你能通过实验验证吗? 是否还可以设计其他的标签集合, 它们能否帮助获得更好的中文分词器?
4. 试比较几种开源分词器在不同语料上的性能（新闻语料，淘宝评论，小说等）。
5. 是否有方法同时进行分词和词性分析？
6. 是否可以用使用 BiLSTM 进行词性分析？与使用 BiLSTM-CRF 算法相比有什么优缺点？



# 参考文献

- [1] 胡壮麟. 语言学教程 (第五版中文本) 清华大学出版社, May 2015 (cited on page 5).
- [2] Dan Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to speech recognition, natural language processing and computational linguistics*. 2nd ed. Pearson, Apr. 2008 (cited on page 6).
- [3] 梁南元. “书面汉语自动分词系统—CDWS”. In: *中文信息学报* 1.2, 46 (1987), p. 46 (cited on pages 13, 14).
- [4] 宗成庆. *统计自然语言处理*. 清华大学出版社, Aug. 2013 (cited on page 14).
- [5] Meishan Zhang, Yue Zhang, and Guohong Fu. “Transition-Based Neural Word Segmentation”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 421–431. doi: [10.18653/v1/P16-1040](https://doi.org/10.18653/v1/P16-1040) (cited on page 29).
- [6] Jie Yang, Yue Zhang, and Shuailong Liang. “Subword Encoding in Lattice LSTM for Chinese Word Segmentation”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019, pp. 2720–2725 (cited on page 29).
- [7] Emma Strubell et al. “Fast and Accurate Entity Recognition with Iterated Dilated Convolutions”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 2670–2680 (cited on page 29).
- [8] Yue Zhang, Qi Liu, and Linfeng Song. “Sentence-State LSTM for Text Representation”. In: *Proceedings of the 56th Annual Meeting of the Association for*

*Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 317–327 (cited on page 29).