Draft

自然语言处理概论

张奇 桂韬 黄萱菁

November 1, 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 语言中的词汇 3

3.2 词形分析 11

3.3 词语切分 17

3.4 词性标注 33

3.5 延伸阅读 44

3.6 习题 45

4 句法分析 **46**

4.1 句法概述 47

4.2 成分句法分析 47

4.3 依存句法分析 47

4.4 句法分析语料库 47

4.5 延伸阅读 47

4.6 习题 47

1 绪论

1.1 拟定章节 1

1.1 拟定章节

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（CNN）

附录 6 递归神经网（RNN+LSTM）
附录 7 Transformer 模型
附录 8 图网络（GCN 等）
附录 9 预训练方法

词法分析 3

3.1 语言中的词汇 . 3
3.2 词形分析 . . . 11
3.3 词语切分 . . . 17
3.4 词性标注 . . . 33
3.5 延伸阅读 . . . 44
3.6 习题 45

词汇是语言知识中的重要环节，在语言学中，**词（Word）**是形式和意义相结合的单位 [1]，也是语言中能够独立运用的最小单位。懂得一个词意味着知道某个的特定读音并与特定语义关联。在书面语中正字法（Orthography）也是词的形式的一种表达。例如：英文单词 “cat’’ 具有语义是 ‘‘猫’’，读音为 “/kæt/’’。由于词是语言运用的基本单位，因此自然语言处理算法中词通常也是基本单元。词的处理也由此成为自然语言处理中重要的底层任务，是句法分析、文本分类、语言模型等任务的基础。

本章首先介绍语言学中词相关的基本概念，在此基础上以介绍词形分析算法，中文分词算法，以及词性分析算法。

3.1 语言中的词汇

词通常是由语素（Morpheme）构成。**语素**是一个语言中意义的最小单元。语素与词不同，语素不能够独立运用而词可以。只包含一个语素的词语称为**简单词（Simple Wold）**，而包含多个语素的词称为**复杂词（Complex Word）**。例如：‘‘电灯’’，包含 ‘‘电’’ 和 ‘‘灯’’ 两个语素此外，根据词在语言中的用途的不同，词还可以被被划分成为**实义词（Content Words）**和**功能词（Function Words）**。实义词包含事物、行为、属性和观念等概念。功能词则是指没有清楚词汇意义或与之有关的明显的概念的词。本节将分别针对语素如何构成词以及如何对词进行分类进行介绍。

语素又称词素

词的形态学

虽然单词的形式和意义之间的关系本质上是任意的，但是由于社会的约定俗成，词的形式具有服从于某种规则的内在结构。在语言学中，研究单词的内部结构和其构成方式的学科称为**形态学（Morphology）**。词是由一个或多个语素构成，语素主要分成两类: **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、中缀、后缀等。例如：

形态学又称为构词学

英语单词 unhappy 中，happy 为原形，-un为前缀
邦托克语单词 fumikas（是强壮的）中，fikas（强壮）为原形，-um-为中缀
俄语单词 barabanshchik（鼓手）中，baraban（鼓）为原形，-shchik为后缀

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’’。

Morphology 本身就是由两个语素构成: morph+ology。后缀-ology 表示 “关于... 的科学”

有些语言的单词通常只包含一个或者两个语素，但是有一些语言的单词则包含多达十个以上的语素。中文中每个单词的语素都很少，也不会根据性、数、格、人称等发生形态变化。但是对于英文，在单词 dog 末尾添加 s 可以将它从单数名词变成复数名词 dogs，对于德语单词 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）。不同语言的词形变化差别非常大，以英语为例，很多英语词都包含两个或两个以上的语素，其词形变化主要有以下几种方式:

- **屈折（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（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：

  组合词也称复合词

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

  根据组合词构成部分之间的语义和语法关系又可以细分为修饰型组合词（Attributive Compounds）、并列型组合词（Coordinative Compounds）以及从属型组合词（Subordinative Compounds）。
- **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
  ```

- **截搭（Blending）**是指将两个词语各自的一部分拼接起来构成新词。例如：

  ```
  smoke（烟） 和 fog（雾） 组合成 smog（烟雾）
  spoon（勺子）和 fork（叉子） 组合成 spork（叉勺）
  ```

- **逆构（Backformation）**是指母语使用者将简单词感知为由多个语素所构

成，将简单词的部分构成新的词汇。例如：

```
edit（编辑，动词） 是由 editor（编辑，名词） 逆构而成
burgle（盗窃，动词）是由 burglar（盗窃，名次）逆构而成
```

editor 和 burglar 历史上都是简单词。

- **缩略（Acronym）**是指短语中多个单词首字母组合在一起构词过程。例如：

```
NLP 代表 Natural Language Processing
IT  代表 Information Technology
```

- **截短（Clipping）**是指将长的单词截为较短的单词。例如：

```
demonstration 简化为 demo
refrigerator 简化为 fridge
```

- **词语新造（Coinage）**是指完全新造一个词语。通常通常包括人名、产品名等。例如：

```
iPhone （手机品牌）
Raspberry Pi （嵌入式开发板卡品牌）
```

通过语素组成词汇也可以反映了语言的一个重要特性：创造性。我们可以理解从未见过的词，也可以通过新颖的方法将语素结合起来创造新词。如果能够自动将词汇分解为语素，可以更好的进行对词汇的进行进一步的分析。

词的词性

词性（Part of Speech，POS）是根据词在句子中扮演的语法角色以及与周围词的关系对词的分类。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。对词性进行划分时通常要综合考虑词的语法特性的各个方面，以某一个标注为主，同时

词性也被称为词类

参照其他标准进行。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[1]，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)。

1: 例如：介词 “in” 后面通常跟名词短语

现在语言学中一个重要的词的分类是区分实义词（Content Words）和功能词（Function Words）。**实义词**表达具体的意义。由于实义词可以不断的增加，因此这类词又被称作**开放类词** (Open class words) 。实义词主要包含名词、动词、形容词等。**功能词**则主要是为了满足语法功能需求。由于功能词相对比较稳定，一个语言中通常很少增加新的功能词，因此功能词又被称作**封闭类词** (Close class words)。功能词主要包含代词、冠词、指示词等。

以英语为例，词性主要包含以下几种:

- **名词（Noun）**是指表示人、物、地点以及抽象概念的一类词。名词按其意义又可以细分为专有名词（Proper noun）和普通名词（Common noun）。普通名词还可以再细分为类名词（Class Noun）、集体名词（Collective Noun）、物质名词（Material Noun）和抽象名词（Abstract Noun）。名词还可以按照其可数性分为可数名词（Countable Noun）和不可数名词（Uncountable Noun）。例如：

  1）专有名词：Shanghai（上海） New York（纽约）
  2）类名词： city（城市） bird（鸟）
  3）集体名词：family（家庭） army（军队）
  4）物质名次：water（水） light（光）
  5）抽象名词：music（音乐） honesty（诚实）

- **动词（Verb）**是指表示动作或状态的一类词，是英语中最复杂的一类词。动词除了具有人称和数的变化之外，还具备一些语法特征，包括：时态（tense）、语态（voice）、语气（mood）、体（aspect）等。动词可以进一步细分为及物动词（Transitive verb）、不及物动词（Intransitive verb）、连系动词（Linking verb）、助动词（Auxiliary verb）、限定动词（Finite verb）、不限定动词（Non-finite verbs）、短语动词（Phrasal verb）等。例如：

1）及物动词： Boys **fly** kites. （男孩们放风筝）
2）不及物动词：Birds **fly**. （鸟会飞）
3）连系动词： The rose **smells** sweet. （玫瑰花香）
4）助动词： I **may** have meet him before.（我以前应该见过他）
5）限定动词： John **reads** papers every day. （约翰每天都读论文）
6）不限定动词：I hope **to see** you this morning.（我希望早上见到你）
7）短语动词： Tom **called up** George.（汤姆给乔治打了电话）

- **形容词（Adjective）**是用来描写或修饰名词的一类词。按照构成，形容词可以被分为简单形容词和复合形容词。按照与其所修饰的名词的关系，形容词还可以被分为限制性形容词（Restrictive adjective）和描述性形容词（Descriptive adjective）。例如：

1）简单形容词：
   a）由一个单词构成 good（好的） long（长的）
   b）由现在分词构成 interesting（令人感兴趣的）
   c）由过去分词构成 learned（博学的）
2）复合形容词： duty-free（免税的） hand-made（手工制作的）
3）限制性形容词：an **Italian** dish （一道意大利菜）
4）描述性形容词：a **delicious** Italian dish （一道美味的意大利菜）

- **副词（Adverb）**是用来修饰动词、形容词、其他副词以及全句的词。按照形式，副词可以被细分为简单副词、复合副词和派生副词。按照意义，副词可以被细分为方式副词、方向副词、时间副词、强调副词等。按照句法作用，可以被分为句子副词、连接副词、关系副词等。例如：

1）简单副词： just（刚刚） only（仅仅）
2）复合副词： somehow（不知怎地） somewhere（在某处）
3）派生副词： interesting → interestingly （有趣地）
4）方式副词： quickly（快地） awkwardly（笨拙地）
5）方向副词： outside（外面） inside（里面）
6）时间副词： recently（最近） always（总是）

7）强调副词： very（很）　　fairly（相当）

- **数词（Numeral）**是表示数目多少或者先后顺序的一类词。表示数目多少的叫做的基数词（Cardinal numeral）。表示顺序先后的叫做序数词（Ordinal numeral）。例如：

  1）基数词： one（1）　　nineteen（19）
  2）序数词： first（第一）　fiftieth（第五十）

- **代词（Pronoun）**是代替名词以及起名词作用的短语、子句和句子的一类词。代词的词义信息较弱，必须通过上下文来确定。代词主要可以细分为人称代词（Personal pronoun）、物主代词（Possessive pronoun）、自身代词（Self pronoun）、相互代词（Reciprocal pronoun）、指示代词（Demonstrative pronoun）、疑问代词（Interrogative pronoun）、关系代词（Relative pronoun）和不定代词（Indefinite pronoun）。例如：

  1）人称代词：
     a）主格：I, you, he, she, it, we, they
     b）宾格：me, you, him, her, it, us, them
  2）物主代词：
     a）形容词性物主代词：my, your, his, her, its, our, their
     b）名词性物主代词：mine, yours, his, hers, its, ours, theirs
  3）自身代词： myself, yourself, himself, herself, itself, ourselves, yourselves, themselves, oneself
  4）相互代词： each other, one another
  5）指示代词： this, that, these, those
  6）疑问代词： who, whom, whose, which, what
  7）关系代词： who, whom, whose, which, that, as
  8）不定代词： some, something, somebody, someone, any, anything, anybody, anyone, no, nothing, nobody, no one

- **冠词（Article）**是置于名词之前，说明名词所指的人或事务的一种功能

词。冠词不能够离开名词而独立存在。英语中冠词有三个冠词：定冠词（Definite article）“the”、不定冠词（Indefinite article）“a/an”和零冠词（Zero article）。

- **介词（Preposition）**是用于表示名词或相当于名词的词语与句中其它词语的关系的一类词。介词在句子中不单独作任何句子成分。介词后面的名词或者相当于名词的词语叫做介词宾语，与介词共同组合成介词短语。从介词的构成来看，其主要包含简单介词（Simple preposition）、复合介词（Compound preposition）、二重介词（Double preposition）、短语介词（Phrasal preposition）、分词介词（Participle preposition）。例如：

介词又称前置词

  1）简单介词： `at, in, of, since`
  2）复合介词： `as for, as to, out of`
  3）二重介词： `from under, from behind`
  4）短语介词： `according to, because of`
  5）分词介词： `including, regarding`

- **连词（Conjunction）**是连接单词、短语、从句或句子的一类词。在句子中也不单独作为句子成分。按照其构成可以细分为简单连词（Simple conjunction）、关联连词（Correlative conjunction）、分词连词（Participial conjunction）、短语连词（Phrasal conjunction）。连词按照其性质可以分为等立连词（co-ordinative conjunction）、从属连词（Subordinative conjunction）。例如：

  1）简单连词： `and, or, but, if`
  2）关联连词： `both ... and, not only ... but also`
  3）分词连词： `supposing, considering`
  4）短语连词： `as if, as long as, in order that`
  5）等立连词： `and, or, but, for`
  6）从属连词： `that, whether, when, because`

- **感叹词（Interjection）**是用来表示喜怒哀乐等情绪或情感的一类词。感叹词也没有实义，也不能在句子中构成任何句子成分，但是与全句有关联。

例如：

Oh, it's you. 啊，是你
Ah, how pitiful! 呀，多可惜!

在语言学研究中，对于词性划分目的、标准、依据等都还存在大量分歧。到目前为止，还没有一个被广泛认可的统一划分标准。在不同的语料集中所采用的划分粒度和标记符号也都不尽相同。宾州大学句法树库（Penn TreeBank）使用了 48 种不同的词性，宾州大学汉语树库（Chinese Penn Treebank）中汉语词性被划分为 33 类，而 Brown 语料库 [2] 中则使用了具有 87 个词性。虽然在语言学中词性还具有很多的仍需要研究的内容，但是由于词性可以提供关于单词和其周边邻近成分的大量有用信息，词性分析也是自然语言处理中重要的基础任务之一。

[2]: Francis (1980), "A tagged corpus–problems and prospects"

3.2 词形分析

词是由语素构成，通过组成词语的语素可以在更好对词汇进行理解和分析。**词形分析 (Morphological Parsing)** 任务就是将一个词分解成为语素的过程。词形分析一个最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 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 [3]：

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

+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

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。

表 3.1: 词形分析输入输出示例

输入	**输出**
cats	cat+N+pl
cat	cat+N+sg
goose	goose+N+sg
walk	walk+N+sg
walk	walk+V
walks	walk+V+3sg
walking	walk+V+prespart
walked	walk+V+past
walked	walk+V+pastpart

表3.1给出了典型词形分析算法的输入和输出结果样例。从结果中可以看到，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。对一个输入串，有限状态机在每个输入字符后进行状态转移。有限状态转换机则是在状态转移的同时给出一个输出。换句话说，有限状态机能够识别一个输入串，有限状态转换机则能够将输入串转换到一个输出串。一个有限状态转换机的可用如下 7 个参数定义:

有限状态转换机又称有限状态转录机、有限状态转换器

- Σ: 输入字符集（有限的字符集合）
- Γ: 输出字符集（有限的字符集合）
- Q: 状态 $q_0, q_1, ...q_N$ 的有限集合
- $q_0 \in Q$: 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 $w \in \Sigma^*$, $\sigma(q, w)$, $\delta(q, w) \subseteq \Gamma^*$, 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_0 为初始状态 q_2 为最终状态。输入符号串和输出符号串中间用冒号分割，$x : y$ 表示当输入串是 x 时，输出 y，x 为 $x : x$ 的简写。

根据自动机理论有限状态转换机与正则关系同构，有限状态转换机存在并运算、差运算、补运算和交运算，以及两个附加的闭包特征。针对基于有限状态转换机词形分析器构造问题，主要需要如下三个运算:

- 交运算 $\cap$: 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。

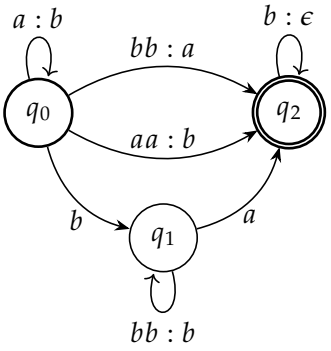

图 3.1: 有限状态转换机 FST 示例

- 取逆 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 $\circ$：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。

将词形分析任务任务转换为一个单词的词汇层和表层的之间的对应。由于英语中很多语素边界发生拼写变化，需要引入正词法规则作为中间层。利用有限状态形态学（Finite-state morphology）范式，该有限状态转换机由三个带子（tape）组成：词汇带子（Lexical tape），中间带子（Middle tape），表层带子（Surface tape），如图3.2所示。针对 foxes，词汇层面表示词形特征（fox+N+pl），中间层面表示词语正词法（fox^#），表层层面表示词语实际拼写（foxes）。

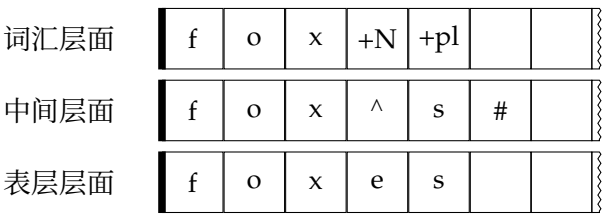

图 3.2: 词汇带子、中间带子和表层带子实例

由字符集 Σ 中的字符构成，表层带子（Surface tape）由字符集 Γ 中字符组成。

以英语中名词的单数、复数屈折变换为例，简要介绍构造有限状态转换机

的过程。为了方便理解，我们首先构造*生成问题*：给定原形和词形特征，输出词汇。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。

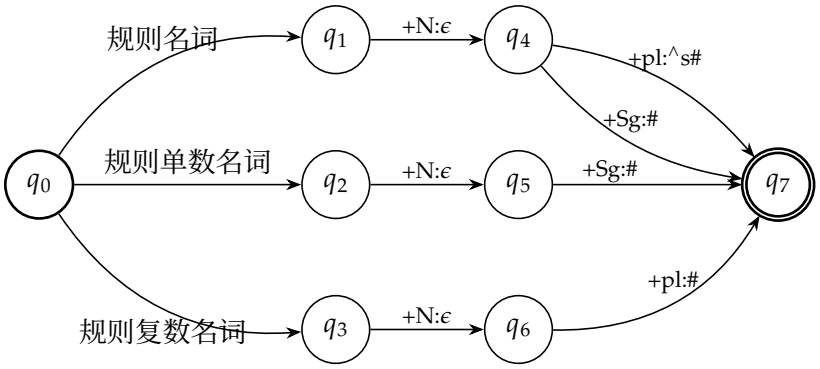

图 3.3: $\text{FST}_{\text{mid}}^{\text{feat}}$ 根据单数复数词形特征生成一种名词的中间表示形式。# 表示单词结尾，语素间用 ^ 分割。

首先，根据3.2中词汇带子和中间带子的定义，构造有限状态转换机 $\text{FST}_{\text{mid}}^{\text{feat}}$，将规则名词单数，不规则名词单数，不规则名词复数分别构造生成路径转换为中间表示，如图3.3中所示。$\text{FST}_{\text{mid}}^{\text{feat}}$ 中 ‘‘规则名词’’，‘‘不规则名词单数’’ 和 ‘‘不规则名词复数’’ 可以进一步根据词典进行展开 (如图3.4)。

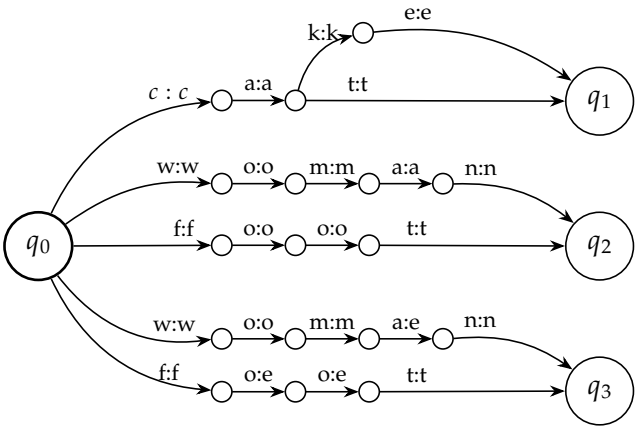

图 3.4: 图3.3中 ‘‘规则名词’’，‘‘不规则名词单数’’，‘‘不规则名词复数’’ 的展开。

通过图3.3和图3.4，可以将词形特征转化成为中间表示。例如:

规则名词单数	“cake+N+Sg” → “cake#”
规则名词复数	“cat+N+pl” → “cat^s#”
不规则名词单数	“foot+N+Sg” → “foot#”
不规则名词复数	“foot+N+pl” → “feet#”

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3.5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^s#’’ 转换成 “boxes#’’, 同时对其他词的拼写不做改动 (“cat^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。

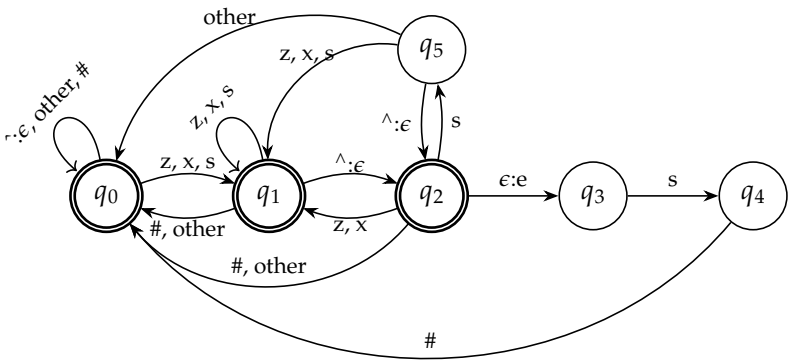

图 3.5: 有限状态转换机实现在 “z, s, x’’ 后插入 e。q_1 表示观察到 “z, s, x’’。在输入语素分隔符 “^’’ 后, 转移到状态 q_2。q_2 实现插入 “e’’, 转移到 q_3, 同时, 当 q_2 输入 “s’’, 后转移到 q_3, 如果此时输入单词解释符 #, 则判断为非法 (即不符合 ‘‘在 z, s, x 后应插入 e’’ 的拼写规则), 并拒绝接收这样的字符串。

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

$$c\text{FST}_{\text{gen}} = \text{FST}_{\text{mid}}^{\text{feat}} \circ (\cap_r \text{FST}_r)$$

同时, 通过取逆操作可以得到词形解析的有限状态转换机:

$$c\text{FST}_{\text{parse}} = \text{FST}_{\text{gen}}^{-1}$$

3.3 词语切分

词是语言中能够独立运用的最小单位，通常也是自然语言处理算法的基础单元，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，词可以较容易的从句子中分割得到。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，而是由一串连续的字符构成。因此，针对汉语等语言的处理算法通常首先需要进行词语切分。

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“自然语言处理概述”一句中的八个汉字对应到四个汉语词汇（“自然”，“语言”，“处理”，“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。

分词规范

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，不可直接用于计算。

北京大学计算语言学研究所俞士汶教授为了构造包含 2600 多万字《人民日报》基本标注语料库，制订了词语切分和词性标注规范 [4] 。针对国家标准分词规范，对分词单位进行了定义和解释。针对人名、地名、机构名、其他专有名词、数词、数量词组、时间词、区别词、述补结构、成语、习用语、非汉字的字符串等情况分别进行了详细的说明。部分标注规范如下所示：

[4]: 俞士汶 et al. (2003), *北大语料库加工规范：切分·词性标注·注音*

1. 人名（nr）：汉族方式的“姓”和“名”单独切分，，“姓”标注为 nrf，“名”标注为 nrg 。例如：李/nrf 明/nrg，欧阳/nrf 洪涛/nrg；
2. 地名（ns）：国名不论长短，作为一个切分单位，地名后有“省”、“市”等单字的现代行政区划名称时，不切分开，如果地名后的行政区划有两个以上的汉字，则将地名同行政区划名称切开。例如：中华人民共和国/ns，上海市/ns，[深圳/ns 特区/n]ns；
3. 机构名（nt）：一般是短语型的，较长，且含有地名或人名等专名，按照 [4] 给出的规范需要先切分，再组合，加方括号标注为 nt。例如：[中国/ns 中文/n 信息/n 学会/n]nt，[复旦/ns 大学/n]nt；
4. 数词与数量词组：基数、序数、小数、分数、百分数一律不予切分，约数，前加副词、形容词或后加“来、多、左右”等助数词的应予切分。例如：一百二十三/m，约/d 一百/m 多/m 万/m；
5. 时间词：年月日时分秒，按年、月、日、时、分、秒切分，“牛年、虎年”等一律不予切分，标注为 t 。例如：2021 年/t 9 月/t 16 日/t，牛年/t；
6. 成语习语：四个字的成语或习用语为一个切分单位，除标注其词类标记 i 或 l 外，还要求根据其在句子中的功能进一步标注子类，超过四个字的成语或习用语，一般不予切分，不分子类。例如：胸有成竹/iv，近水楼台先得月/i；

需要注意的是，不同的分词规范之间也存在一定的不同，微软亚洲研究院黄昌宁教授 [5] 所给出的分词标注规范中有不少与《北京大学语料库加工规范》存在不同。例如，在 [5] 中姓名需要整体标出，含有外文和数字的命名实体应整体一起标注等。此外，虽然标注规范中尽可能的给出了详尽的细节，但是其中还存在一些弹性，由于中文词汇本身具有开放性和动态性，不同人之间也存在

[5]: 黄昌宁 et al. (2006), *中文文本标注规范 (5.0 版)*

认同差异，通用分词标准也是中文分词的难题。

歧义切分

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”，但是也可能会被切分为“南京 | 市长 | 江 | 大桥”。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 如果汉字串 AJB 中，AJ、JB 都可以分别组成词汇，则汉字串 AJB 被称为交集型切分歧义，此时汉字串 J 称作交集串。交集型切分歧义也被称为偶发歧义，当两个有交集的词“偶然”的相邻出现时这样的歧义才会发生。

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 如果汉字串 AB 满足 A，B，AB 同时为词，则汉字串 AB 被称为组合型切分歧义。组合性切分歧义也称为固有歧义，组合歧义的是词固有的属性，不依赖于“偶然”发生的上下文。

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 如果汉字串 ABC 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，则汉字串 ABC 被称为真歧义。

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。

上述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 [6] 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。

[6]: 梁南元 (1987), “书面汉语自动分词系统—CDWS”

未登录词识别

未登录词（Out Of Vocabulary，OOV）是指在训练语料中没有出现或者词典当中没有，但是在测试数据中出现的词。根据分词算法所采用的技术不同，未登录词所代表的含义也稍有区别。基于词典的分词方法所指的未登录词就是所依赖的词典中没有的单词。对于完全基于统计方法不依赖词典特征的方法，未登录词则是指训练语料中没有出现的单词。而对于融合词典特征的统计方法，未登录词则是指训练语料和词典中均未出现的词。

未登录词又称生词（Unknown Words）

汉语具有很强的灵活性，未登录词的类型也十分复杂，可以粗略的将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)：
  ①人名（如：杰辛达，周杰伦）；
  ②地名（例如：新江湾，张江）；
  ③组织机构名（例如：亚洲善待博士组织，中央第四巡视组）；
  ④时间和数字（例如：2021-09-16，正月初四，110 亿人民币）；
- 专业名词：出现在专业领域的新词 (例如: 图灵机，新冠病毒，埃博拉);
- 其他专有名词：新出现的产品名、电影名、书籍名等。

针对中文分词中歧义切分和未登录词造成的损失情况，黄昌宁教授和赵海教授在 Bakeoff 一 2003 的四个中文分词语料库中针对的当年最好的算法进行了测试和统计，结果标明未登录词造成的分词精度失落比歧义切分造成的精度失落至少大 10 倍左右 [7] 。宗成庆教授在新闻领域的语料也进行了类似的统计实验，结果发现未登录词造成的分词错误超过 98%，其中由命名实体引起的分词错误占到了 55% 左右 [8] 。由此可见，未登录词是中文分词的一个主要瓶颈。

[7]: 黄昌宁 et al. (2007), "中文分词十年回顾"

[8]: 宗成庆 (2013), *统计自然语言处理*

中文分词任务定义

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 3.2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3.2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 3.2: 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，这些算法试图根据词典，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 $c[i:i+L-1], c[i:i+L-2], \ldots, c[i:i]$，其中 $c[i:j] \triangleq c_i c_{i+1} \ldots c_j$ 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，针对句子 ‘‘今晚的长安街流光溢彩’’，前向最大分词的过程如表3.2所示（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’’, ‘‘。’’}）

表 3.3: 表3.2中句子的前向最大匹配分词过程。eos 表示句子结尾。

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，。	。

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3.2的所对应的序列编码如表3.4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 3.4: 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:

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

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3.5列出了中文分词任务常用的模板。

此外，在模板设计时还可以加入字符的类别（例如：阿拉伯数字、中文数字、标点符号、英文字母等）以及字典信息（例如：x_{i-1}, x_i 是否是词典中的二字

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 3.5: 中文分词常见模板。“例子”一栏中包含对应模板在表3.2中句子的第 5 个位置时的取值。

词，x_i, x_{i+1} 是否是词典中某个词的开头等)。基于字的分词方法可以有效的平衡训练语料中出现的词语和未登录词，并且可以使用模板特征引入词典信息。相较于基于词典的方法，基于字的分词方法通常也可以省略未登录词的识别模块。

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 是可能的原始输入句子集合，Y 是可能的句子切分集合，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 $\overline{\alpha} \in \mathbb{R}^d$ 间的点积构成，a_i 代表第 i 个特征的参数:

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

对于参数 $\overline{\alpha}$，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 $\overline{\alpha}$。

Algorithm 1: 基于感知机算法的 Score 函数训练算法

输入: 训练数据 (x_i, y_i)
输出: $\overline{\alpha}$
初始化参数 $\overline{\alpha} = 0$
for $t = 1$ to T **do**
  for $i = 1$ to N **do**
    $z_i = \arg\max_{y \in GEN(x_i)} \Phi(y) \cdot \overline{\alpha}$
    if $z_i \neq y_i$ **then**
      $\overline{\alpha} = \overline{\alpha} + \Phi(y_i) - \Phi(z_i)$
    end if
  end for
end for
return $\overline{\alpha}$

感知机算法所需的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。Y. Zhang 和 S. Clark 在其论文中所使用的具体特征模板如下表所示 [9]：

表 3.6: 输入特征模板

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 $c1$ 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

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。针对于这一问题，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集中选取前 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 可以理解为一种 ''松驰'' 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。

Algorithm 2: Beam Search 解码算法

输入: 原始句子 s
输出: src
初始化 $src = [[]], tgt = []$
for $index = 0$ to s.length $- 1$ **do**
    var $char = s[index]$
    for $item$ in src **do**
        var $item_1 = item$
        $item_1$.append($char$.toWord())
        tgt.insert($item_1$)
        if $item.length > 1$ **then**
            var $item2 = item$
            $item2$[$item2$.length - 1].append($char$)
            tgt.insert($item_2$)
        end if
    end for
    $src = tgt$
    $tgt = []$
end for

用一个例子来理解分词中的 Beam Search 算法：假设有这样一句话“今晚的长安街流光溢彩。”，Beam 大小为 2，在解码到第 5 个字之前的候选集中将会有两个候选分词结果:

“今晚/的/长安街”
“今晚/的/长安/街”

对于第六字“流”可以扩展出 4 个新的候选分词句：

“今晚/的/长安街/流”
“今晚/的/长安街流”
“今晚/的/长安/街流”
“今晚/的/长安/街/流”

经过打分排序后的结果为：

“今晚/的/长安街/流”
“今晚/的/长安/街流”
“今晚/的/长安街流”
“今晚/的/长安/街/流”

由于 Beam 大小设置为 2，因此只取头两句句子继续解码，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。

基于双向长短期记忆网络结合条件随机场的分词

随着深度学习技术的发展，很多中文分词算法也采用了基于神经网络模型。循环神经网络（Recurrent Neural Network, RNN）相较于前馈神经网络等要求固定输入长度的神经网络结构，更适用于处理长度不固定的序列数据。特别符合文本、语音等在内的数据特性，广泛应用于自然语言处理任务的很多任务中。长短期记忆网络（LSTM）[10, 11] 是循环神经网络的一个变体，可以在一定程度上缓解简单循环神经网络的梯度消失和梯度爆炸问题。

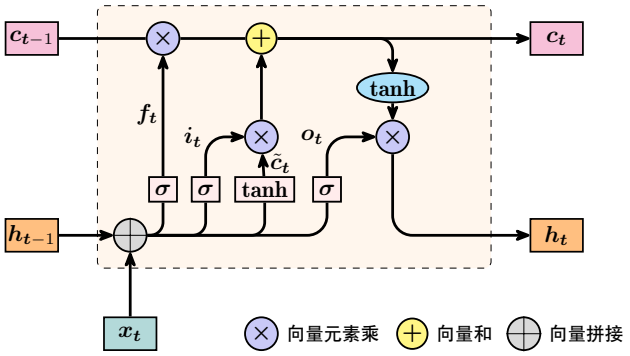

图 3.6: LSTM 网络的循环单元结构

LSTM 网络循环单元结构如图3.6所示。LSTM 网络引入了新的内部状态（internal state）$\mathbf{c_t} \in \mathbb{R}^D$，专门用来进行信息传递。此外，LSTM 网络还引入了门控机制（Gating Mechanism）来控制信息传递路径。通过**遗忘门** $\mathbf{f_t}$ 控制上一个时刻的内部状态 $\mathbf{c_{t-1}}$ 需要遗忘多少信息。**输入门** $\mathbf{i_t}$ 用来控制当前时刻的候选状态 $\mathbf{c_t}$ 有多少信息需要保存。**输出门** $\mathbf{o_t}$ 控制当前时刻内部状态 $\mathbf{c_t}$ 有多少信息需要输出给外部状态 $\mathbf{h_t}$。三个门的计算方式为：

$$\mathbf{i_t} = \sigma(\mathbf{W_i x_i} + \mathbf{U_i h_{t-1}} + \mathbf{b_i}) \tag{3.1}$$

$$\mathbf{f_t} = \sigma(\mathbf{W_f x_i} + \mathbf{U_f h_{t-1}} + \mathbf{b_f}) \tag{3.2}$$

$$\mathbf{o_t} = \sigma(\mathbf{W_o x_i} + \mathbf{U_o h_{t-1}} + \mathbf{b_o}) \tag{3.3}$$

其中 $\sigma(\cdot)$ 为 Logistic 函数。候选状态 $\tilde{\mathbf{c}}_t$、内部状态 $\mathbf{c_t}$ 以及隐藏输出 $\mathbf{h_t}$ 通过如下公式计算：

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W_c x_t} + \mathbf{U_c h_{t-1}} + \mathbf{b_c}) \tag{3.4}$$

$$\mathbf{c_t} = \mathbf{f_t} \otimes \mathbf{c_{t-1}} + \mathbf{i_t} \otimes \tilde{\mathbf{c}}_t \tag{3.5}$$

$$\mathbf{h_t} = \mathbf{o_t} \otimes \tanh(\mathbf{c_t}) \tag{3.6}$$

更为详细的介绍请参与邱锡鹏教授《神经网络与深度学习》的第六章 [12]。

在自然语言处理的很多任务中，一个时刻的输出不但与过去某个时刻的信息相关，也与后续时刻的信息相关。**双向长短期记忆网络**（Bidirectional LSTM，BiLSTM）是用来建模上述问题的一种方法。BiLSTM 是由两层长短期记忆网络组成，它们结构相同但是信息传递的方向不同。双向长短期记忆网络还可以结合条件随机场，更有效的利用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3.7给出了一个使用双向长短期记忆网络结合条件随机场（BiLSTM+CRF）进行分词的框架。

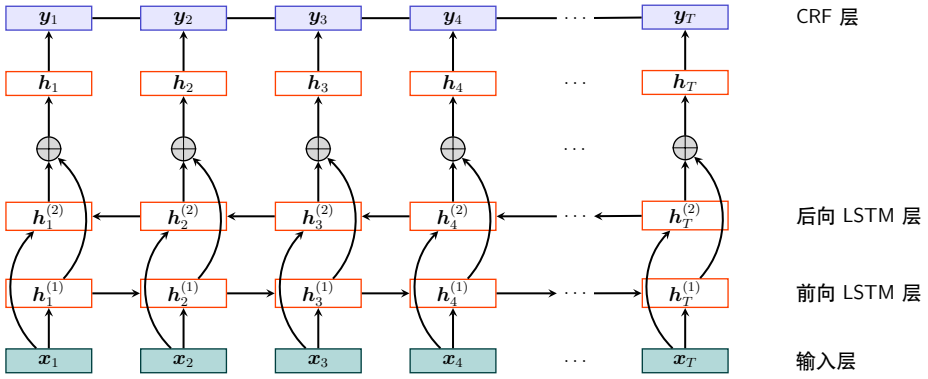

图 3.7: 基于 BiLSTM+CRF 的神经网络分词模型

在基于神经网络的分词算法中，通常采用与基于字的统计方法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，对于给定一个中文句子 $\boldsymbol{x} = \{c_1, c_2, \cdots, c_T\}$，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），输出标签序列 $\boldsymbol{y} = \{y_1, y_2, \cdots, y_T\}$。具体模型如图3.7所示，BiLSTM-CRF 主要包含三层：输入层、双向长短期记忆网络层和 CRF 层。在输入层，需要将每个字转换为低维稠密的**字向量**（Character Embedding）$\boldsymbol{x}_i$。

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 $(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T)$ 输入到双向 LSTM 各个时间步，再将正向 LSTM 输出的隐状态序列 $(\mathbf{h}_1^{(1)}, \mathbf{h}_2^{(1)}, ..., \mathbf{h}_T^{(1)})$ 与反向 LSTM 隐状态序列的 $(\mathbf{h}_1^{(1)}, \mathbf{h}_2^{(2)}, ..., \mathbf{h}_T^{(2)})$。在此基础上按位置进行拼接 $\mathbf{h}_i = \mathbf{h}_i^{(1)} \oplus \mathbf{h}_i^{(2)}$，从而得到完整的隐状态序列。

对于给定的长度为 T 的输入 $\left[x\right]_1^T$，定义网络的输出矩阵为 $f_\theta\left(\left[x\right]_1^T\right)$（简写为 f_θ），其中 $\left[f_\theta\right]_{i,t}$ 表示参数为 θ 的网络对于句子 $\left[x\right]_1^T$ 的第 t 个单词的第 i 标签的打分。同时定义转移值矩阵 A，其中 $\left[A\right]_{i,j}$ 为相邻的两个单词的标签从 i 标签到第 j 标签的值，$\left[A\right]_{i,0}$ 为开始标签为第 i 标签的值。由于转移值矩阵也是模型参数的一部分，因此整个模型的参数 $\tilde{\theta} = \theta \cup \{[A]_{i,j} \quad \forall i,j\}$。对于输入句 $\left[x\right]_1^T$ 的某个特定标签序列 $\left[i\right]_1^T$ 定义为转移值和网络值的和，具体公式如下：

$$s\left(\left[x\right]_1^T, \left[i\right]_1^T, \tilde{\theta}\right) = \sum_{t=1}^{T}\left(\left[A\right]_{[i]_{t-1},[i]_t} + \left[f_\theta\right]_{[i]_t,t}\right) \tag{3.7}$$

通过 softmax 函数可以将某个标签序列的得分根据所有可能标签序列 $\left[j\right]_1^T$ 的得分进行归一化，得到标签序列的条件概率：

$$P\left([i]_i^T \mid [x]_1^T, \tilde{\theta}\right) = \frac{e^{s\left([x]_1^T,[i]_1^T,\tilde{\theta}\right)}}{\sum_{\forall [j]_1^T} e^{s\left([x]_1^T,[j]_1^T,\tilde{\theta}\right)}} \tag{3.8}$$

由此可以进一步得到对于输入 $\left[x\right]_1^T$ 的正确标签序列 $\left[y\right]_1^T$ 的条件概率的对数似然（log-likelihood）：

$$\log P\left([y]_i^T \mid [x]_1^T, \tilde{\theta}\right) = s\left(\left[x\right]_1^T, \left[y\right]_1^T, \tilde{\theta}\right) - \log\left(\sum_{\forall [j]_1^T} e^{s\left([x]_1^T,[j]_1^T,\tilde{\theta}\right)}\right) \tag{3.9}$$

基于最大化对数似然目标，以及公式3.9的线性计算方法 [13, 14]，可以根据标注语料训练得到模型参数 $\tilde{\theta}$。根据模型参数，使用维特比（Viterbi）算法可以对任意句子预测每个字的标签序列，从而得到分词结果。

中文分词语料库

如前文所述，现代分词系统的训练通常常需要依赖大规模标注语料。本节将介绍目前较为广泛使用的部分中文分词语料库。

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！

同时他们还行制定了《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 3.7: 中文分词语料库汇总

3.4 词性标注

词性是词语的基本属性，根据其在句子中所扮演的语法角色以及与周围词的关系进行分类。**词性标注**是指在给定的语境中确定句子中各词的词性 [15] 。词性标注是句法分析的基础，也是自然语言处理中一项重要的基础任务。

[15]: 吴立德 (1997), *大规模中文文本处理*

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具有不同的词性。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所对应的词性。另一方面，具有兼类词多为常用词，而且越是常用词，其用法就越多。英语 “like” 就具有动词、名词、介词等多种词性。针对北京大学计算语言学研究 200 万字语料库统计，发现兼类词所占比例仅有 11%，但是出现的次数缺占到了 47%[16]。对 Brown 语料库的统计也发现超过 80% 的词通常只有一个词性。

具有多个词性的词语称为兼类词

此外，由于在语言学研究中，还没有一个被广泛认可的统一词性划分标准，在不同的语料集中所采用的划分粒度和标记符号也都不尽相同，这也在一定程度上对词性标注问题研究造成了困难。表3.8列出了在宾州树库 (PTB) 中所使用的词性。而宾州大学汉语树库（Chinese Penn Treebank）中汉语词性被划分为 33 类，北京大学计算语言学研究所给出的语料库加工规范中包含 26 个基础词性，74 个扩展词性。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。本节中将主要以 PTB 标准为例。

标签	描述	标签	描述
CC	并列连词	CD	数字
DT	限定词	EX	*there*
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	to	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

表 3.8: 宾州树库中的词性标签

基于规则的词性标注

基于规则的词性标注算法是最早应用于词性标注任务的一类方法，其核心思想是利用词典和搭配规则针对词语和上下文进行分析，从而得到句子中每个词语的词性的方法。早期通常采用人工的方法来构建规则，随着机器学习算法的不断发展以及资源的不断完善，也出现了一些基于机器学习方法的规则自动学习算法。在本节中我们将重点介绍基于转换的 Brill Tagger 方法 [17]。

Brill Tagger 是一种利用错误驱动方法学习转换规则的词性标注算法。在 Brown 语料库上仅使用 71 个规则就得到接近 95% 的分析准确率。其分析算法的主要过程如下：

1. **初始化**：对于词典中包含的词语，根据词语最常使用的词性设置初始值；对于词典中没有的单词根据词性分析结果设置初始值（例如：以

大写字母开头的设置为专有名词)。

2. **规则转换**：根据补丁规则对初始标注进行转换，补丁规则包含以下三类：

   a) 如果某单词词性为 a，并且其所在上下文为 C，那么将其词性转换为 b；

   b) 如果某单词词性为 a，并且其具有词汇属性 P，那么将其词性转换为 b；

   c) 如果某单词词性为 a，并且其周边范围 R 内有一个词汇具有属性 P，那么将其词性转换为 b；

例如：补丁规则 **"NN VB PREV-TAG"** 表示，如果一个单词被标注为了 NN（名词），并且它前前面的单词标注为了 TO（不定式 "to"），那么将这个单词的词性转换为 VB（动词）。可以用用于解决类似 "to **book** a hotel" 中对于单词 book 的词性默认标注错误的问题。

Brill Tagger 中对于补丁规则的学习方法采用了基于错误驱动的有监督模板学习方法。首先根据现有的初始词典和补丁模板针对从训练语料中预留的用于获取模板的语料进行分析，将错误的分析结果汇总为三元组 $<tag_a, tag_b, num>$ 形式，表示一个单词的词性应该为 tag_b，但是在评测语料中有 num 次都被标注了为了词性 tag_a。根据所得到的三元组，利用以下模板生成补丁规则：

- 前一个（或者后一个）单词被标注为了 z
- 前面第二个（或者后面第二个）单词被标注为了 z
- 前面两个（或者后面两个）单词某一个被标注为了 z
- 前面三个（或者后面三个）单词某一个被标注为了 z
- 前一个单词被标注为了 z，并且后一个单词被标注为了 w
- 前一个单词被标注为了 z，并且前面第二个（或者后面第二个）单词被标注为了 w
- 当前单词是（不是）首字母大写
- 前一个单词是（不是）首字母大写

根据每个 $<tag_a, tag_b, num>$ 三元组，以及利用上述模板得到的补丁规则，可以计算利用该规则可以修复的错误标记数，以及利用该规则所引入的新的错误数。根据上述数值，选择改进最大的补丁规则加入规则列表中，并进行新一轮的分析和规则生成。

基于错误驱动的规则学法方法可以在一定程度上缓解人工规则抽取上的时间成本和人力成本。在词性标注问题中取得了不错的效果。但是其效果严重依赖于训练语料的规模和质量，同时也较难处理未登录词。此外，受到规则模板复杂度的限制，其效果通常也低于基于统计机器学习的方法。

基于隐马尔科夫模型的词性标注

隐马尔科夫模型又称隐马尔可夫模型

隐马尔科夫模型（Hidden Markov Model，HMM）是马尔科夫过程扩充而来的一种随机过程，其基本理论是由数学家 Baum 及其同事构建并逐步完善。随着隐马尔科夫模型在语音识别领域 [18] 取得巨大成功，其在自然语言处理众多序列标注任务中也得到了广泛应用并取得了非常好的效果。一个隐马尔科夫模型可用如下 5 个参数定义:

- N: 状态数。所有的状态记为 $S = \{s_1, s_2, ..., s_N\}$。系统在 t 时刻的状态记为 q_t。$Q = \{q_1, q_2, ..., q_T\}$，为长度为 T 的状态序列。
- M: 观察值数。所有的可能观察值记为 $V = \{v_1, v_2, ...v_M\}$。系统在 t 时刻的观测值记为 o_t。$O = \{o_1, o_2, ..., o_T\}$，为长度为 T 的观测序列。
- π：初始状态概率。$\pi = [\pi_i]_{1\times N}, \pi_i = P(q_1 = s_i), 1 \le i \le N$，表示初始时刻 $t = 1$ 时处于某个状态 s_i 的概率。
- A: 状态转移概率矩阵。$A = [a_{ij}]_{N\times N}, a_{ij} = P(q_{t+1} = s_j | q_t = s_i), 1 \le i, j \le N$，表示在时刻 t 处于状态 S_i 的条件下，下一时刻 $t + 1$ 转移到状态 s_j 的概率。
- B: 观测概率矩阵。$B = [b_j(k)]_{N\times M}$，$b_j(k) = P(o_t = v_k | q_t = s_j), 1 \le j \le N, 1 \le k \le M$，表示在时刻 t 处于状态 s_j 的条件下，观测到 v_k 的概率。

为了简化起见，隐马尔科夫模型可以表示成 $\lambda = (A, B, \pi)$。M，N 也隐含的已经包含在 A, B, π 中。隐马尔科夫模型的三个主要问题是：

问题 1：观测概率计算 在给定模型 $\lambda = (A, B, \pi)$ 的情况下，如何根据观测序列 $O = \{o_1, o_2, ..., o_T\}$ 计算 $P(O|\lambda)$，即在给定模型情况下，如何观测序列的概率。

问题 2：状态序列预测 在给定模型 $\lambda = (A, B, \pi)$ 和观测序列 $O = \{o_1, o_2, ..., o_T\}$ 计算 $P(O|\lambda)$ 的情况下，如何得到与该观测序列最匹配的状态序列 $Q = \{q_1, q_2, ..., q_T\}$，即如何根据观测序列推断出隐藏的状态序列。

问题 3：模型参数学习 在给定观测序列 $O = \{o_1, o_2, ..., o_T\}$ 情况下，如何调整模型参数 $\lambda = (A, B, \pi)$ 使得该序列的 $P(O|\lambda)$ 最大，即如何训练模型使其能最好的建模观测序列。

关于问题 1，问题 2 以及问题 3 的求解方法可以参阅李航博士《统计学习方法（第二版）》第 10 章中的相关内容 [19]。

针对词性标注任务，使用隐马尔科夫模型可以按照如下方式构建和学习模型。N 为词性数，$S = \{s_1, s_2, ..., s_N\}$ 为词性表，包含所使用到的所有词性信息。M 为单词数，$V = \{v_1, v_2, ...v_M\}$ 为单词词表，包含所有单词。给定一个由 T 个单词组成的句子 $W = w_1, w_2, ..., w_T$，即相当于观测序列 $O = \{o_1, o_2, ..., o_T\}$，$o_i$ 为句子中第 i 个单词 w_i。状态序列 $Q = \{q_1, q_2, ..., q_T\}$ 则表示输入句子中单词对应的词性。根据训练语料, 可以使用最大似然估计的 Baum-Welch 方法高效的得到模型参数。在此基础上，针对输入的句子可以利用维特比（Viterbi）算法应用动态规划求解状态路径，从而得到对应的词性。图3.8给出了基于词性标注的隐马尔科夫模型概率图模型样例。

在实际应用过程中，使用隐马尔科夫模进行词性标注，通常还需要解决两个问题：长句子和未登录词。在《人民日报》语料库中，有些句子非常长，甚至会超过 120 个字。虽然这种长句子在真实环境中很少出现，但是对于模型的设计和实现都带来了一定挑战。因此，通常会限定一个句子中单词的最大数量。如果一个句子超过了所设定的最大长度，则寻找距离最大长度最近的标点，并

加西亚·马尔克斯所著的魔幻现实主义小说《族长的秋天》中，很多句子“一逗到底”，超过 1000 个字

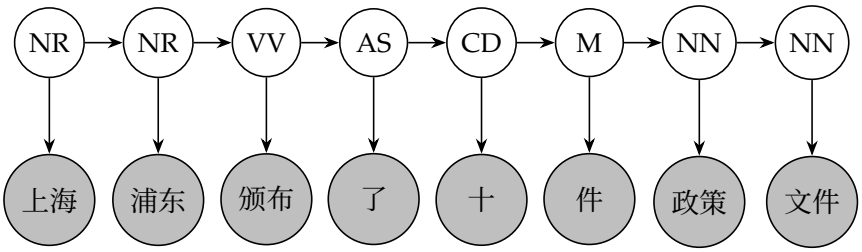

图 3.8: 词性标注隐马尔科夫模型概率图模型样例。

在标点处将句子截断。对于词典当中没有出现的未登录词，由于观测概率矩阵 B 中不存在，也需要进行特殊处理。第一种做法是在单词表中增加一个“未登录词”项，同时在观测概率矩阵中设置该词以同样的概率观察到所有标记类别。这种做法较为粗糙，在本章中我们介绍过词的一个重要分类角度是开放类词和封闭类词。未登录词通常属于名词、动词、形容词等开放类词语。其中人名、地名、机构名等名词又以占据了很大的比例。因此第二种做法是引入词法规则，对人名、地名、数词、副词等进行判断。此外，还可以根据更大规模的统计未登录词的词性，从而设定更合理的观测概率。

基于卷积神经网络的词性标注

在深度神经网络应用于自然语言处理任务之前，绝大多数自然语言处理算法依赖于特征工程。Collobert 等人 [14] 在 2011 年所提出的“从零开始的 NLP”框架利用统一的具有多个隐藏层的神经网络解决了多个自然语言处理中任务，省去了特征工程的步骤，推动了深度学习在自然语言处理任务中的快速发展。在本节我们以词性标注任务为例介绍该方法。

[14]: Collobert et al. (2011), “Natural language processing (almost) from scratch”

如图3.9所示，基于卷积神经网络的词性标注模型首先通过表查询 (Lookup Table)$LT_W(\cdot)$ 将单词通过表查询将单词转换为词性向量表示，词向量的维度是 d_{wrd}。

$$LT_W(w) = \left\langle W \right\rangle_w^1 \tag{3.10}$$

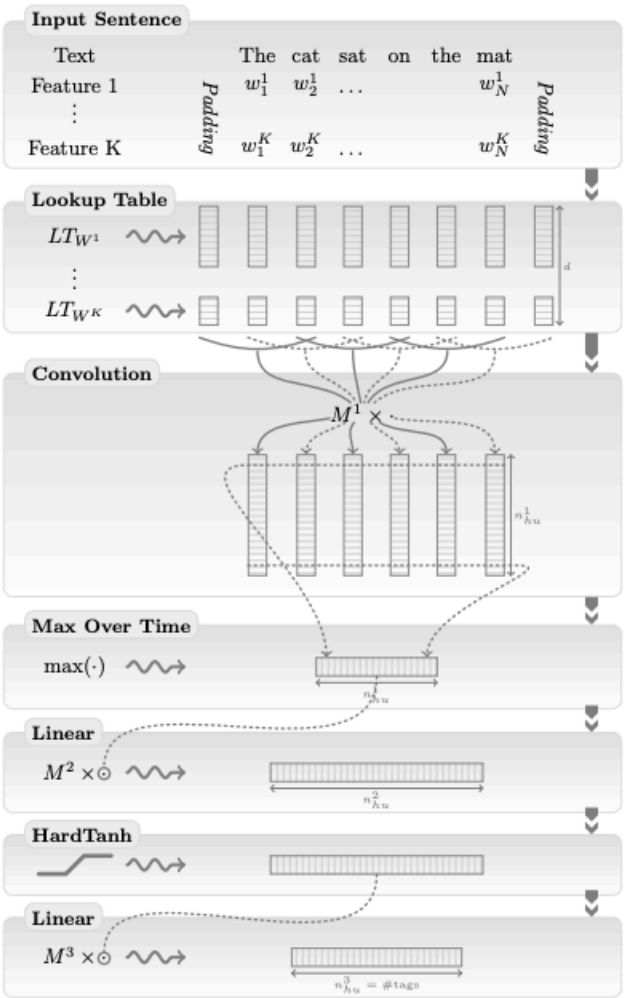

图 3.9: 基于卷积神经网的词性标注模型结构

其中 $W \in \mathbb{R}^{d_{wrd} \times |D|}$，D 是包含有限个单词的的字典，$\langle W \rangle_w^1 \in \mathbb{R}^{d_{wrd}}$ 表示 W 矩阵的第 w 列。W 矩阵也是要进行学习的参数。对于给定的任意一句包含 T 个单子的句子 $[w]_1^t$，通过表查询层对序列中的每个单词进行转换，得到如下表查询层输出矩阵：

$$LT_W([w]_1^T) = \left(\langle W \rangle_{[w]_1}^1, \langle W \rangle_{[w]_2}^1 ... \langle W \rangle_{[w]_T}^1 \right) \tag{3.11}$$

除了单词本身之外，中还可以提供一些其他特征，例如该单词在词典中最常见词性等信息。因此，可以将单词更一般的表示为 K 个离散特征 $w = \mathcal{D}^1 \times \mathcal{D}^2 \times ... \mathcal{D}^K$，$\mathcal{D}^K$ 是第 k 维特征的字典。$LT_{W^k}(\cdot)$ 是每维特征的查询表，$W^k \in \mathbb{R}^{d^k_{wrd} \times |D^k|}$ 是第 k 维特征的嵌入向量矩阵，$d^k_{wrd} \in \mathbb{N}$ 是用户给定的向量维度。对于一个单词 w，其特征向量的维度 $d_{wrd} = \sum_k d^k_{wrd}$，通过表查询得到连接后的向量：

$$LT_{W^1,...,W^K}(w) = \begin{pmatrix} LT_{W^1}(w_1) \\ \vdots \\ LT_{W^K}(w_K) \end{pmatrix} = \begin{pmatrix} \langle W^1 \rangle^1_{w_1} \\ \vdots \\ \langle W^K \rangle^1_{w_K} \end{pmatrix} \tag{3.12}$$

由此，可以得到如下表查询层输出矩阵：

$$LT_{W^1,...,W^K}([w]^T_1) = \begin{pmatrix} \langle W^1 \rangle^1_{[w_1]_1} & \dots & \langle W^1 \rangle^1_{[w_1]_T} \\ \vdots & & \vdots \\ \langle W^K \rangle^1_{[w_K]_1} & \dots & \langle W^K \rangle^1_{[w_K]_T} \end{pmatrix} \tag{3.13}$$

在表查询层后连接的是卷积层（Convolutional Layer），根据所设置的窗口大小 d_{win}，将每个单词周边的单词拼接起来构成具有 $d_{wrd} d_{win}$ 维度的向量：

$$f^1_\theta = \langle LT_W([w]^T_1) \rangle^{d_{win}}_t = \begin{pmatrix} \langle W \rangle^1_{[w]_{t-d_{win}/2}} \\ \vdots \\ \langle W \rangle^1_{[w]_t} \\ \vdots \\ \langle W \rangle^1_{[w]_{t+d_{win}/2}} \end{pmatrix} \tag{3.14}$$

f^1_θ 会被给入单层或者多层的卷积层，第 l 层的第 t 列向量可以根据如下公式计算得到：

$$\langle f^l_\theta \rangle^1_t = W^l \langle f^{l-1}_\theta \rangle^{d_{win}}_t + b^l \qquad \forall t \tag{3.15}$$

在同一层中 W^l 为相同参数。对于 f_θ^l 中每一维在公式3.15计算完成后，都要进行非线性变化，可以采用如下方式：

$$\left[f_\theta^l\right]_i = HardTanh\left(\left[f_\theta^l\right]_i\right), \tag{3.16}$$

$$HardTanh(x) = \begin{cases} -1 & \text{if } x < -1 \\ x & \text{if } -1 \le x \le 1 \\ - & \text{if } x > 1 \end{cases} \tag{3.17}$$

通过公式3.15得到特征向量更多了反映了局部特征，并且数量与句子长度相关。为了得到全局特征并且维度固定的特征向量，需要引入池化层（Pooling Layer），在这里使用的是随时间推移最大化（Max Over Time）方法。给定通过卷积层计算得到的矩阵 f_θ^{l-1}，池化层输出的向量 f_θ^l 计算如下：

$$\left[f_\theta^l\right]_i = \max_x \left[f_\theta^{l-1}\right]_{i,t} \qquad 1 \le i \le n_{hu}^{l-1} \tag{3.18}$$

针对通过池化层计算得到的向量 f_θ^l 需要继续进行线性变换，再利用公式3.16进行非线性变换，之后再叠加新的线性层后完成特征提取工作。线性变换层对于输入 f_θ^{l-1} 利用如下公式计算得到其输出 f_θ^l:

$$f_\theta^l = W^l f_\theta^{l-1} + b^l \tag{3.19}$$

在分类阶段，采用句子级别对数似然方法（Sentence-Level Log-Likelihood）。除了网络输出矩阵 $f_\theta\left(\left[x\right]_1^T\right)$（简写为 f_θ）之外，引入转移值矩阵 A，对于输入句 $\left[x\right]_1^T$ 的某个特定标签序列 $\left[i\right]_1^T$ 定义为转移值和网络值的和。基于最大化对数似然目标，可以根据标注语料训练得到模型参数 $\tilde{\theta}$。根据模型参数，使用维特比（Viterbi）算法可以获得任意句子中每个词的词性。

相关算法以及公式在第3.3节基于 BiLSTM-CRF 方法进行分词部分进行了详细介绍，也可参考文献 [14] 查看详细算法和公式推导。

词性标注语料库

从前面几节的介绍可以知道，词性标注算法的训练过程都依赖标注语料集合。对不同算法的效果进行对比也依赖于标准测试集合。本节将介绍几种常见的包含词性标签的语料库。

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 [20] 。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

[20]: Petrov et al. (2011), "A universal part-of-speech tagset"

(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/

表 3.9: 词性标注语料库汇总

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

3.5 延伸阅读

关于中文分词，我们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 [21]，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 [22]，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 [23]）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 [24], 同时引入全局结点来弥补上下

文建模的不足，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。

3.6 习题

1. 语言学中词和语素的定义分别是什么？其主要的不同是什么？
2. 如何处理词性标注算法中的未登录词？
3. 如何同时进行分词和词性标注？
4. 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。
5. 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？
6. 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）。
7. 是否可以用使用 BiLSTM 或者 BiLSTM-CRF 进行词性分析？与使用 BiLSTM-CRF 算法相比有什么优缺点？

4 句法分析

4.1 句法概述 . . . 47
4.2 成分句法分析 47
4.3 依存句法分析 47
4.4 句法分析语料库 47
4.5 延伸阅读 . . . 47
4.6 习题 47

词汇是语言知识中的重要环节，在语言学中，**词（Word）**是形式和意义相结合的单位 [1]，也是语言中能够独立运用的最小单位。懂得一个词意味着知道某个的特定读音并与特定语义关联。在书面语中正字法（Orthography）也是词的形式的一种表达。例如：英文单词 “cat’’ 具有语义是 ‘‘猫’’，读音为 “/kæt/’’。由于词是语言运用的基本单位，因此自然语言处理算法中词通常也是基本单元。词的处理也由此成为自然语言处理中重要的底层任务，是句法分析、文本分类、语言模型等任务的基础。

本章首先介绍语言学中词相关的基本概念，在此基础上以介绍词形分析算法，中文分词算法，以及词性分析算法。

4.1 句法概述

4.2 成分句法分析

基于上下文无关文法的成分句法分析

基于概率上下文无关文法的成分句法分析

4.3 依存句法分析

基于图的依存句法分析

基于神经图的依存句法分析

基于转移的依存句法分析

基于神经转移的依存句法分析

4.4 句法分析语料库

4.5 延伸阅读

4.6 习题

参考文献

Here are the references in citation order.

[1] Victoria Fromkin, Robert Rodman, and Nina Hyams. *An introduction to language*. Cengage Learning, 2018 (cited on pages 3, 46).

[2] W Nelson Francis. "A tagged corpus–problems and prospects". In: *Studies in English linguistics for Randolph Quirk* (1980), pp. 192–209 (cited on page 11).

[3] Dan Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to speechrecognition, natural language processing and computational linguistics*. 2nd ed. Pearson, Apr. 2008 (cited on page 12).

[4] 俞士汶 et al. *北大语料库加工规范：切分·词性标注·注音*. 北京大学计算语言学研究所, 2003 (cited on page 18).

[5] 黄昌宁, 李玉梅, and 朱晓丹. *中文文本标注规范 (5.0 版)*. 微软亚洲研究院, 2006 (cited on page 18).

[6] 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 page 20).

[7] 黄昌宁 and 赵海. "中文分词十年回顾 ". In: *中文信息学报* 21.3 (2007), pp. 8–19 (cited on page 21).

[8] 宗成庆. *统计自然语言处理*. 清华大学出版社, Aug. 2013 (cited on page 21).

[9] Yue Zhang and Stephen Clark. "Chinese segmentation with a word-based perceptron algorithm". In: *Proceedings of the 45th annual meeting of the association of computational linguistics*. 2007, pp. 840–847 (cited on page 25).

[10] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780 (cited on page 27).

[11] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM". In: *Neural computation* 12.10 (2000), pp. 2451–2471 (cited on page 27).

[12] 邱锡鹏. *神经网络与深度学习*. 北京: 机械工业出版社, 2020 (cited on page 28).

[13] Lawrence R Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286 (cited on page 30).

[14] Ronan Collobert et al. "Natural language processing (almost) from scratch". In: *Journal of machine learning research* 12.ARTICLE (2011), pp. 2493–2537 (cited on pages 30, 38, 41).

[15] 吴立德. *大规模中文文本处理*. 复旦大学出版社, 1997 (cited on page 33).

[16] 张虎, 郑家恒, and 刘江. "语料库词性标注一致性检查方法研究 ". In: *中文信息学报* 18.5 (2004), pp. 12–17 (cited on page 33).

[17] Eric Brill. "A Simple Rule-Based Part of Speech Tagger". In: *Proceedings of the Third Conference on Applied Natural Language Processing*. ANLC '92. Trento, Italy: Association for Computational Linguistics, 1992, pp. 152–155. DOI: 10.3115/974499.974526 (cited on page 34).

[18] Lalit Bahl et al. "Maximum mutual information estimation of hidden Markov model parameters for speech recognition". In: *ICASSP'86. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 11. IEEE. 1986, pp. 49–52 (cited on page 36).

[19] 李航. *统计学习方法(第二版)*.清华大学出版社, May 2019 (cited on page 37).

[20] Slav Petrov, Dipanjan Das, and Ryan McDonald. "A universal part-of-speech tagset". In: *arXiv preprint arXiv:1104.2086* (2011) (cited on page 43).

[21] 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 (cited on page 44).

[22] 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 44).

[23] 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 44).

[24] 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 44).