自然语言处理导论

张奇 桂韬 黄萱菁

2023 年 1 月 14 日

.

数学符号

数与数组

α	标量
$\boldsymbol{\alpha}$	向量
$\boldsymbol{A}$	矩阵
$\mathbf{A}$	张量
$\boldsymbol{I}_n$	n 行 n 列单位矩阵
$\boldsymbol{v}_w$	单词 w 的分布式向量表示
$\boldsymbol{e}_w$	单词 w 的独热向量表示：[0,0,...,1,0,...0]，w 下标处元素为 1

索引

α_i	向量 $\boldsymbol{\alpha}$ 中索引 i 处的元素
α_{-i}	向量 $\boldsymbol{\alpha}$ 中除索引 i 之外的元素
$w_{i:j}$	序列 w 中从第 i 个元素到第 j 个元素组成的片段或子序列
A_{ij}	矩阵 $\boldsymbol{A}$ 中第 i 行、第 j 列处的元素
$\boldsymbol{A}_{i:}$	矩阵 $\boldsymbol{A}$ 中第 i 行
$\boldsymbol{A}_{:j}$	矩阵 $\boldsymbol{A}$ 中第 j 列
$\boldsymbol{A}_{ijk}$	三维张量 $\mathbf{A}$ 中索引为 (i,j,k) 处元素
$\mathbf{A}_{::i}$	三维张量 $\mathbf{A}$ 中的一个二维切片

集合

$\mathbb{A}$	集合
$\mathbb{R}$	实数集合
$0, 1$	含 0 和 1 的二值集合
$0, 1, ..., n$	含 0 和 n 的正整数的集合
$[a, b]$	a 到 b 的实数闭区间
$(a, b]$	a 到 b 的实数左开右闭区间

线性代数

$\boldsymbol{A}^\top$	矩阵 $\boldsymbol{A}$ 的转置
$\boldsymbol{A} \odot \boldsymbol{B}$	矩阵 $\boldsymbol{A}$ 与矩阵 $\boldsymbol{B}$ 的 Hardamard 乘积
det$\boldsymbol{A}^\top$	矩阵 $\boldsymbol{A}$ 的行列式
$[\boldsymbol{x}; \boldsymbol{y}]$	向量 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 的拼接
$[\boldsymbol{U}; \boldsymbol{V}]$	矩阵 $\boldsymbol{A}$ 与 $\boldsymbol{V}$ 沿行向量拼接
$\boldsymbol{x} \cdot \boldsymbol{y}$ 或 $\boldsymbol{x}^\top \boldsymbol{y}$	向量 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 的点积

微积分

$\frac{\mathrm{d}y}{\mathrm{d}x}$	y 对 x 的导数
$\frac{\partial y}{\partial x}$	y 对 x 的偏导数
$\bigtriangledown_{\boldsymbol{x}} y$	y 对向量 $\boldsymbol{x}$ 的梯度
$\bigtriangledown_{\boldsymbol{X}} y$	y 对矩阵 $\boldsymbol{X}$ 的梯度
$\bigtriangledown_{\mathbf{X}} y$	y 对张量 $\mathbf{X}$ 的梯度

概率与信息论

$a \perp b$	随机变量 a 与 b 独立
$a \perp b \mid c$	随机变量 a 与 b 关于 c 条件独立
$P(a)$	离散变量概率分布
$p(a)$	连续变量概率分布
$a \sim P$	随机变量 a 服从分布 P
$\mathbb{E}_{x \sim P}[f(x)]$ 或 $\mathbb{E}[f(x)]$	$f(x)$ 在分布 $P(x)$ 下的期望
Var$(f(x))$	$f(x)$ 在分布 $P(x)$ 下的方差
Cov$(f(x), g(x))$	$f(x)$ 与 $g(x)$ 在分布 $P(x)$ 下的协方差
$H(f(x))$	随机变量 x 的信息熵
$D_{KL}(P \parallel Q)$	概率分布 P 与 Q 的 KL 散度
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	均值为 $\boldsymbol{\mu}$、协方差为 $\boldsymbol{\Sigma}$ 的高斯分布

数据与概率分布

$\mathbb{X}$	数据集
$\boldsymbol{x}^{(i)}$	数据集中第 i 个样本（输入）
$\boldsymbol{y}^{(i)}$ 或 $y^{(i)}$	第 i 个样本 $\boldsymbol{x}^{(i)}$ 的标签（输出）

函数

$f : \mathcal{A} \longrightarrow \mathcal{B}$	由定义域 $\mathcal{A}$ 到值域 $\mathcal{B}$ 的函数（映射）f
$f \circ g$	f 与 g 的复合函数
$f(\boldsymbol{x}; \boldsymbol{\theta})$	由参数 $\boldsymbol{\theta}$ 定义的关于 $\boldsymbol{x}$ 的函数（也可以直接写作 $f(\boldsymbol{x})$，省略 $\boldsymbol{\theta}$）
$\log x$	x 的自然对数函数
$\sigma(x)$	Sigmoid 函数 $\dfrac{1}{1+exp(-x)}$
$\|\|\boldsymbol{x}\|\|_p$	$\boldsymbol{x}$ 的 L^p 范数
$\|\|\boldsymbol{x}\|\|$	$\boldsymbol{x}$ 的 L^2 范数
$\mathbf{1}^{\text{condition}}$	条件指示函数：如果 condition 为真，则值为 1；否则值为 0

本书中常用写法

- 给定词表 $\mathbb{V}$，其大小为 $|\mathbb{V}|$
- 序列 $x = x_1, x_2, ..., x_n$ 中第 i 个单词 x_i 的词向量 $\boldsymbol{v}_{x_i}$
- 损失函数 $\mathcal{L}$ 为负对数似然函数：$\mathcal{L}(\boldsymbol{\theta}) = -\sum_{(x,y)} \log P(y|x_1...x_n)$
- 算法的空间复杂度为 $\mathcal{O}(mn)$

目 录

2. 词汇分析

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这个词的读音以及其语义。在书面语中，正字法（Orthography）也是词形式的一种表达。例如：英文单词 “cat” 具有的语义是 “猫”，读音为 “/kæt/”。由于词是语言运用的基本单位，自然语言处理算法中词通常也是基本单元。因此，词的处理也是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语规范化相关算法，中文分词算法，以及词性分析算法。

2.1 语言中的词汇

词通常是由语素（Morpheme）构成。**语素**又称词素，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 word），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。

2.1.1 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）, 又称构词学。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，是指能在字典中查的到的语素，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：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

Morphology 本身就是由两个语素构成：morph+ology，后缀-ology 表示 “关于... 的科学”。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语素，但是有一些语言的单词则包含多达十个以上的语素。汉语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。但是对于英语单词 dog，在末尾添加 s 可以将它从单数名词变成复数名词 dogs。对于德语单词 bäcker，在末尾添加 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，其词形变化如表2.1所示。

表 2.1 英语中常见词形变换

词形变化	说明	举例
屈折 Inflection	通过 “词根 + 词缀” 的方式构成和原形 “同一类型” 的词	名词后加 -s 后缀复数名词（cat+s） 动词后加 -ed 后缀动词的过去式（walk+ed）
派生 Derivation	通过 “词根 + 词缀” 的方式构成和原形 “不同类型” 的词	employ 添加后缀 -ee 变为 employee meaning 添加后缀 -less 变为 meaningless
复合 Compounding	通过组合多个词根构成一个新词，也称组合词	home + work → homework water + proof → waterproof
附着 Cliticization	通过 ‘词根 + 附着语” 的方式 “附着” 在词根上	I’m 中的’m 代表 am 附着在 I 上 We’re 中’re 代表 are
截搭 Blending	两个词语各自的一部分拼接起来构成新词	smoke（烟）+ fog（雾）→ smog（烟雾） spoon（勺子）+ fork（叉子）→ spork（叉勺）
缩略 Acronym	短语中多个单词首字母组合在一起组合成词	NLP 代表 Natural Language Processing IT 代表 Information Technology
截短 Clipping	将长的单词截为较短的单词	demonstration 简化为 demo refrigerator 简化为 fridge

通过语素组成词汇也可以反映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，可以更好地对词汇进行进一步的分析。

2.1.2 词的词性

词性（Part of Speech，POS）也称词类，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，例如：介词 “in” 后面通常跟名词短语。通过词性可以为语法分析、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)。

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地增加，因此这类词又被称作**开类词** (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**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）物质名词：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：

例如：1) 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：

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：

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（迅速）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。英语中冠词有三种冠词：定冠词（Definite article）“the”、不定冠词（Indefinite article）“a/an”和零冠词（Zero article）。

介词（Preposition）又称前置词，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：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并列连词（Coordinative 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，对于词性划分的标准、依据甚至目的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。英语宾州树库（Penn TreeBank）使用了 48 种不同的词性，汉语宾州树库（Chinese Penn Treebank）中汉语词性被划分为 33 类，而布朗语料库（Brown Corpus）[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。

2.2 词语规范化

在对自然语言文本进行分析前，通常需要对文本进行规范化的处理。文本的规范化处理主要包含句子切分、词语切分、词语规范化等步骤。由于绝大部分语言的句子结束符数量有限，符号歧义性相对容易处理，因此句子切分可以通过词典结合模板或者有监督分类方法都可以达到较高的准确率。**词语规范化**（Word Normalization）任务是将单词或词形转化为标准形式，针对有多种形式的单词使用一种单一的形式进行表示。本章中主要讨论词语的规范化问题，包括词语切分、词形分析和词干提取。

2.2.1 词语切分

对于绝大部分的印欧语系的语言来说，词语之间通常由分隔符区分开来。英语是印欧语系（Indo-European languages）的典型代表，英语句子中绝大部分单词之间都由空格或标点分割。但是以汉语为代表的汉藏语系（Sino-Tibetan languages）的语言中，单词之间通常没有分隔符。因此在对文本进行分析前，通常需要将句子切分为单词序列，称之为**词语切分**（Word Tokenization）。

词语切分任务可以定义为：给定一个符号串 $x = c_1, c_2, \cdots, c_n$，（其中 c_i 对于英文来说是字母、数字、标点符号等，对于中文来说是汉字、数字、标点符号等），输出是一个词形（Token）序列 $y = t_1, t_2, \cdots, t_m$，可能会省略或删除其中的部分标点符号。例如：

输入：Let’s first understand what’s NLP.

输出：`Let␣'s␣first␣understand␣what␣'s␣NLP␣.`

通过上面的例子可以看到虽然英语句子中绝大部分的单词可以通过以空格和标点符号为分隔符进行识别，但是还是存在一些例外情况，例如：缩写（Prof.），日期（02/18/2022），数字（562,000），连字符（upper-case）等。此外，还可以看到“Let’s”被划分为了“Let”和“’s”。正是因此，在词语切分的定义中使用了词形。**词形**（Token）指的是在一个特定文档中的某个能够表达语义含义的字符序列。虽然在大部分情况下词形和单词没有区别，对于某些场景和算法有必要对单词和词形进行区分。

在英语中，一些特殊的符号和数字也需要完整的保留到一起。比如数字（“67.20”）、时间（22:37）、微博话题标签（# 北京 2022 年冬奥会 #）、Email 地址（cs_nlp@fudan.edu.cn）等。在特定的应用中有时也会将“Hong Kong”，“Head, Shoulders, Knees and Toes”划分为一个词形。这也使得在某些应用中词语切分与命名实体识别任务（将在第 7 章信息抽取中进行详细介绍）紧密相关。

通常情况下针对英语等印欧语系语言的词语切分任务可以采用基于有限状态自动机（Finite State Automata）融合正则表达式的方法完成。但是针对汉语、日语、阿拉伯语等词语中间没有分隔符的语言，词语切分问题更加复杂，在后序章节中我们将以中文分词为例进行详细介绍。

2.2.2 词形还原

词形还原（Lemmatization）是将词的各种变化形式还原其词根的过程。通过词形还原可以实现词语的规范化，单词的不同变化形式统一为词根。

例如：原始输入句：They are working on interesting tasks

词形还原后：they be work on interesting task

词形还原可以通过词形分析 (Morphological Parsing) 完成。**词形分析**是将一个词分解成为语素的过程。最简单的方法是词典查表法，讲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，构造包含所有绝大多数词形的词典能够有效地支撑许多应用场景。由于用词方式变化和新词的不断出现，需要对这个字典进行及时维护。但是，对于某些语言（特别是土耳其语、阿拉伯语等黏着语系的语言）枚举所有词的词形变换则是不可能的。

例如：土耳其语词汇 uygarlaştıramadıklarımızdanmışsınızcasına 是由以下 10 项变换组合而成[3]：

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

可以看到，在一些语言中由于词形变换的复杂性，一个词的原形可能衍生出很多不同的词。采用词典匹配的方法很难达到较好的分析效果。因此，需要更有效率的词形分析算法。典型的词形分析算法包括基于有限状态转换机（Finite State Transducer）方法，融合词典和有限状态转换机的方法以及统计机器学习方法等。

2.2.3 词干提取

词干提取 (Stemming）是词形分析的简化版本，其目标是将具有词形变化（通常是屈折或派生）的词语还原为其词干（Word Stem）。与词形分析不同，词干提取并不要求还原的词干一定与其语言学词根完全一致，只需要将相关的单词映射为统一的词干。甚至词干本身可能并不是一个单词。例如：词干提取算法 Porter Stemmer [4] 将 argue, argued, argues, arguing, 以及 argus 都转换为 argu.

最简单的词干提取算法可以通过查询词表的方法获得，这种方法依赖词典所能覆盖的单词数

量，并且需要及时更新以应对不断出现的新词。另外一种常见的算法是后缀剥离（Suffix-stripping），通过定义一组规则，将特定的后缀从词形中删除。

例如：如果单词以'ed'结尾，则删除'ed'
如果单词以'ing'结尾，则删除'ing'
如果单词以'ly'结尾，则删除'ly'

后缀剥离方法虽然可以很好的处理词语的规则变形，但是无法处理特殊变形（如：ran，took 等）。后缀替代（Suffix Substitution）算法可以在一定程度上解决上述问题。与后缀剥离不同，后缀替代是定义规则将单词后缀替换为另外一个后缀。

例如：如果单词以'ational'结尾，则替换为'ate'（relational → relate）
如果单词以'ing'结尾，则替换为'ϵ'（working → work）
如果单词以'zzes'结尾，则替换为'Z'（quizzes → quiz）

Porter Stemmer 就采用了这种后缀替换的方法进行词干提取。

2.3 中文分词

以英语为代表的印欧语系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，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。

2.3.1 中文分词概述

中文分词（Chinese Word Segmentation, CWS）是指将连续字序列转换为对应的词序列的过程，也可以看做在输入的序列中添加空格或其他边界标记的过程。中文分词任务可以形式化表示为：给定中文句子 $c_1, c_2, \cdots, c_n$，其中 c_i 为单个字符，输出词序列 $w_1, w_2, \cdots, w_m$，其中 w_j 是中文单词。

例如：复旦大学是中国人自主创办的第一所高等院校。
分词结果：复旦大学 | 是 | 中国人 | 自主 | 创办 | 的 | 第一 | 所 | 高等 | 院校 |。

由于汉语中语素绝大部分是单个汉字，很多情况下单独使用时是词，不单独使用时又是构词成分，这使得汉语构词具有很大的灵活性和很强的组词能力。对于新概念的表示不需要创造新的汉字，仅需使用现有汉字就可以灵活地创造新词。但是，这正是因为汉语的这些特点，中文分词任务面临了巨大的挑战，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。

1. 分词规范

汉语中对词的具体界定目前还没有定论。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[5]。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：

(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,按照参考文献 yu2003cwsstandard 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[6] 所给出的分词标注规范与《北京大学语料库加工规范》存在很多不同。例如，微软亚洲研究院给出的规范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[6]。但是《北京大学语料库加工规范》中姓名要分为姓和名两个词。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，不同人之间也存在认同差异，通用分词标准也是中文分词的难题。

2. 切分歧义

由于汉语构词方式的灵活性，使得同一个汉语句子很可能产生多个不同的分词结果，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

切分方式 1：南京市 | 长江大桥

切分方式 2：南京 | 市长 | 江大桥

该例句中“南京”、“南京市”、“市长”、“长江”都是词语，因此同样一个句子可以出现多种切分方式。

这种切分歧义在汉语中普遍存在。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。

交集型切分歧义是指汉字串 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。

切分方式 1：乒乓 | 球 | 拍卖 | 完 | 了 |。

切分方式 2：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。

组合型切分歧义是指如果汉字串 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。

切分方式 1：他 | 马上 | 过来 | 。

切分方式 2：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。

真歧义是指如果汉字串 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。

切分方式 1：白天 | 鹅 | 在 | 水 | 里 | 游泳 | 。

切分方式 2：白天鹅 | 在 | 水 | 里 | 游泳 | 。

对这个句子来说，以上两种切分方式在语法和语义上都是正确的，需要考虑句子上下文环境，甚至是篇章内容才能进行正确判断。

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，许多歧义切分通常真实的上下文环境通常不能成立。例如, “平淡”根据定义属于组合型切分歧义, 但实际上“平 | 淡”这样的切分方式能够符合上下文语境的情况非常罕见。根据刘开瑛教授在《中文文本自动分词和标注》中的给出的统计，汉语新闻文本中每 1000 个词约出现 16 次交集型切分歧义[7]。

3. 未登录词识别

未登录词（Out Of Vocabulary，OOV）又称生词（Unknown Words），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是指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。

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类型：

- 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：中国中文信息学会，复旦大学）；
  ④时间和数字（例如：2021-09-16，正月初四，110 亿人民币）；
- 专业名词：出现在专业领域的词语 (例如: 图灵机，偶氮二甲酸二乙酯，胞质溶胶);
- 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-2003 的四个中文分词语料库，针对当年最好的多种中文分词算法进行了统计，结果均标明未登录词造成的分词精度损失比歧义切分造成的精度损失至少大 10 倍左右[8]。宗成庆教授在新闻领域的语料也进行了类似的实验，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[9]。由此可见，未登录词是中文分词的一个重要难题。

2.3.2 基于最大匹配的中文分词

最大匹配（Maximum Matching）分词算法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类。这些算法试图根据给定的词典，利用贪心搜索策略找到分词方案。

前向最大匹配算法的基本思想是，从左向右扫描句子，选择当前位置与词典中最长的词进行匹配，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 “他是研究生物化学的一位科学家”，前向最大分词的过程如表2.2所示。为简单起见，词典中词语最大长度假设为 4，词表为 {“他”，“是”, “研究”, “生物化学”, “的”, “一”, “位”, “科学家”}。

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结果作为最终的输出（例如选择平均词长较大的一个）。

可以看到，基于词典的分词方法具有简单、快速、可控、仅依赖词表等优点。但对于没有在

表 2.2 前向最大匹配分词过程示例

时间步	开始位置	候选匹配	输出
1	1	他是研究，他是研，他是，他	他
2	2	是研究生，是研究，是研，是	是
3	3	研究生物，研究生，研究，研	研究
4	5	生物化学，生物化，生物, 生	生物化学
5	9	的一位科，的一位，的一, 的	的
6	10	一位科学，一位科，一位, 一	一
7	11	位科学家，位科学，位科, 位	位
8	12	科学家，科学, 科	科学家

词典中出现的词没有很好的处理方案，同时对于分词歧义的处理能力也不足。

2.3.3 基于线性链条件随机场的中文分词

根据中文分词任务定义，我们可以将分词过程看做是对于字的分类。具体来说，对于输入句子中的每一个字 c_i，根据它在分词结果中的位置赋予不同的标签。可以假设一个字在词语中有四个位置：开始（B）、中间（I）、结尾（E）以及单独成词（S）。

例如：输入句子：他是研究生物化学的一位科学家。
分词结果：他 | 是 | 研究 | 生物化学 | 的 | 一 | 位 | 科学家 |。
对应标记：他/S 是/S 研/B 究/E 生/B 物/I 化/I 学/E 的/S 一/B 位/E 科/B 学/I 家/E 。/S

这里的“字”不仅包含汉字, 还包含英文字母、数字、标点符号等所有可能出现在汉语文本中的符号。

通过 BIES 标签可以将分词问题转换为字的分类问题。此外，由于一个字 c_i 的分类结果与其周边的字的分类结果有关联。比如 c_i 被分类为 B 标签表示一个单词的开头时，c_{i+1} 标签就不应该分类为 S 标签表示一个成词的字。因此，中文分词任务也是典型的序列标注问题。可以采用条件随机场等结构化机器学习方法进行解决。

条件随机场（Conditional Random Field，CRF）试图对多个变量在给定观测值后的条件概率行建模。$x = \{x_l, x_2, \ldots, x_n\}$ 为观测序列，$y = \{y_l, y_2, \ldots, y_n\}$ 为对应的标记序列，条件随机场的目标是构建条件概率模型 $P(y|x)$。在中文分词任务中，观察序列 x 对应输入的字序列 $\{c_1, c_2, \cdots, c_n\}$，标记序列为每个字对应的 BIES 标签。在实际应用中，对序列任务进行建模时，通常使用如图2.1所示的链式结构，即线性链条件随机场（Linear-chain CRF）。

条件随机场使用势函数和图结构上的团来定义条件概率 $P(y|x)$。给定观测序列 x，线性链式条件随机场主要包含两种关于标记变量的团：单个标记变量 y_i 和相邻的标记变量 y_{i-1}, y_i。选用

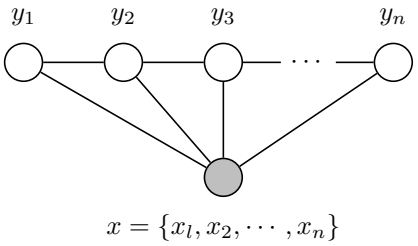

图 2.1 线性链条件随机场结构图

指数势函数并引入特征函数 (Feature Function)，条件概率则定义为：

$$P(y|x) = \frac{1}{Z(x)} \exp\left(\sum_j \sum_{i=2}^{n} \lambda_j t_j(x, y_i, y_{i-1}, i) + \sum_k \sum_{i=1}^{n} \mu_k s_k(x, y_i, i)\right) \tag{2.1}$$

$$Z(x) = \sum_y \exp\left(\sum_j \sum_{i=2}^{n} \lambda_j t_j(x, y_i, y_{i-1}, i) + \sum_k \sum_{i=1}^{n} \mu_k s_k(x, y_i, i)\right) \tag{2.2}$$

其中 $t_j(x, y_i, y_{i-1}, i)$ 是转移特征函数（Transition feature function），用于刻画相邻标记之间的相关关系以及测序列对它们的影响；$s(x, y_i, i)$ 是状态特征函数（Status feature function），用于刻画观测序列对标记变量的影响；λ_j 和 μ_k 为参数；$Z(x)$ 为规范化因子，在所有可能的输出序列上进行求和，用于确保公式2.1是正确定义的概率。通常转移特征函数 t_j 和状态特征函数 s_k 的取值为 0 或 1，当满足特征条件时取值为 1，否则为 0。线性链式条件随机场完全由特征函数 t_j、s_k 以及其对应的参数 λ_j 和 μ_k 决定。

针对中文分词任务，典型的转移特征如下：

$$t_j(x, y_i, y_{i-1}, i) = \begin{cases} 1 & \text{if } x_i = \text{“复”} \text{ 并且 } y_i = \text{“B”} \text{ 并且 } y_{i-1} = \text{“E”} \\ 0 & \text{otherwise} \end{cases}$$

表示第 i 个观测值为“复”时，相应的标记 y_i 和 y_{i-1} 很可能分别为 B 和 E。典型的状态特征如下：

$$s_j(x, y_i, i) = \begin{cases} 1 & \text{if } x_i = \text{“上”} \text{ 并且 } y_i = \text{“B”} \\ 0 & \text{otherwise} \end{cases}$$

表示第 i 个观测值为“上”时，相应的标记 y_i 很可能为 B。

如何设计有效的特征函数对于序列标注任务是至关重要的。针对中文分词问题，可以使用模板的方式从当前字的上下文中构建。表2.3列出了中文分词任务常用的模板。其中 T(c) 表示字符 c

的类型，包括阿拉伯数字、中文数字、标点符号、英文字母等。基于特征模板和训练语料，可以自动生成转移特征以及状态特征。

表 2.3 基于线性链条件随机场的中文分词常见模板

模板名	描述
$c_k(k=-2,-1,0,1,2)$	c_{i-2}、c_{i-1}、c_i、c_{i+1}、c_{i+2} 位置字符
$c_kc_{k+1}(k=-2,-1,0,1)$	$c_{i-2}c_{i-1}$、$c_{i-1}c_i$、c_ic_{i+1}、$c_{i+1}c_{i+2}$ 双字组
$c_{-1}c_1$	$c_{i-1}c_{i+1}$ 双字组
$\mathrm{T}(c_k)(k=-2,-1,0,1,2)$	$\mathrm{T}(c_{i-2})$、$\mathrm{T}(c_{i-1})$、$\mathrm{T}(c_i)$、$\mathrm{T}(c_{i+1})$、$\mathrm{T}(c_{i+2})$ 位置字符类型

基于线性链条件随机场中文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，基于线性链条件随机场中文分词方法通常可以省略未登录词的识别模块。关于线性链条件随机场的模型训练和预测算法可以参阅李航博士《统计学习方法（第二版）》第 11 章中的相关内容[10]。

2.3.4 基于感知器的中文分词

中文分词可以定义为将连续字序列转换为对应的词序列的过程。使用 $x=\{c_1,c_2,\ldots,c_n\}$ 表示输入字序列，$y=\{w_1,w_2,\ldots,w_m\}$ 表示输出词序列，$\mathrm{F}(x)$ 表示最优分词结果。根据上述定义中文分词可以形式化的表达为：

$$\mathrm{F}(x)=\underset{y\in\mathrm{GEN}(x)}{\arg\max}\,\mathrm{SCORE}(y) \tag{2.3}$$

其中 $\mathrm{GEN}(x)$ 代表对于每一个输入句子 x 可能的所有候选输出，$\mathrm{SCORE}(y)$ 为针对分词结果 y 评分函数。

基于感知器中文分词方法，将每一个分词后的单词序列 y 定义一个特征向量 $\Phi(x,y)\in\mathbb{R}^d$，其中 d 代表模型中的特征数量，评分函数 $\mathrm{SCORE}(y)$ 定义为由向量 $\Phi(x,y)$ 与参数 $\boldsymbol{\alpha}\in\mathbb{R}^d$ 间的点积：

$$\mathrm{SCORE}(y)=\Phi(x,y)\cdot\boldsymbol{\alpha} \tag{2.4}$$

将中文分词任务转化为上述问题后，需要解决如下三个问题：

问题 1：词序列预测 在给定模型参数 $\boldsymbol{\alpha}$ 和输入字序列 $x=\{c_1,c_2,\ldots,c_n\}$ 的情况下，如何得到最优的词序列 $y=\{w_1,w_2,\ldots,w_m\}$，即模型解码算法。

给定模型参数情况下，对输入句子的词序列预测问题，根据公式2.3的定义需要计算所有候选分词结果得分。但是，每一个句子都有数量十分庞大的候选分词结果，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集束搜索（Beam Search）算法进行解码。集束搜索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。集束搜索可以理解为一种“松驰”过的贪心算法，它并不能保证一定会得到得分最高的候选序列。算法2.1给出了应用于基于感知器中文分词集束搜索算法详细流程。

代码 2.1: 基于感知器中文分词解码算法

输入: 待分词汉字序列 $x = \{c_1, c_2, \ldots, c_n\}$
输出: 分词结果 $y = \{w_1, w_2, \ldots, w_m\}$
src =[[]], tgt=[]; // 初始化;
for $i = 1$ **to** n **do**
  foreach $item \in src$ **do**
    item1 = c_i; // 当前字作为新词的开始;
    item2 = item[item.length]+c_i; // 当前字附加到 item 最后一个候选词上;
    tgt 中添加 item1 和 item2 ;
  end
  使用评分函数 SCORE 对 tgt 中所有分词结果进行打分;
  对 tgt 中的评分结果进行排序，保留前 K 个;
  src = tgt ;
  tgt =[];
end
return src[1] ; // 返回 src 中最好结果

基本思路是：针对输入的句子 x，解码器每次读入一个字 c_i，根据候选词队列，采用两种方法扩充候选结果集：1）作为下一个词的开始；2）添加到上一个候选词的末尾。对现有的候选分词结果进行评分，保留得分最高的前 K 个候选分词结果。重复上述过程，直到句子结束，输出得分最高的分词结果。

问题 2：模型参数学习 在给定训练语料 $\{x_i, y_i\}$ 的情况下，如何调整模型参数 $\boldsymbol{\alpha}$，使其能针对训练语料得到最好的分词结果，即模型参数学习算法。

对于模型参数 $\boldsymbol{\alpha}$ 的学习问题，可以使用感知器算法进行训练。对训练语料中每一个句子，根据现有模型参数进行解码得到分词结果，与正确答案进行比对，如果结果错误则更新参数 $\boldsymbol{\alpha}$。算法对整个训练语料迭代 T 轮。算法2.2给出了训练算法的详细流程。也可以采用平均感知器（Average perceptron）算法避免训练过程中的过拟合问题。

问题 3：特征定义 给定输入字序列 x 和分词后的词序列 y，如何定义特征对词序列进行描述，并能够区分分词序列的优劣，即构建特征向量 $\Phi(x, y) \in \mathbb{R}^d$。

针对特征向量 $\Phi(x, y)$ 的定义问题，感知器算法所需的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。在训练和解码时会使用特征模板将解码得到的序列映射到特征向量。

代码 2.2: 基于感知器算法的评分函数训练算法

输入: 训练数据 $D = (x_i, y_i)$
输出: 模型参数 $\boldsymbol{\alpha}$
for $i = 1$ **to** T **do** // T 轮迭代;
  foreach $(x, y) \in D$ **do**
    $\boldsymbol{z} = \arg\max_{y \in \text{GEN}(x)} \text{SCORE}(y)$; // 使用算法 2.1 给出的解码算法;
    if $\boldsymbol{z} \neq y$ **then**
      $\boldsymbol{\alpha} = \boldsymbol{\alpha} + \Phi(x, y) - \Phi(x, \boldsymbol{z})$;
    end
  end
end
return $\boldsymbol{\alpha}$

Zhang 和 Clark 在其论文中所使用的具体特征模板[11] 如表2.4所示。

表 2.4 基于感知器的中文分词算法所使用特征模板样例[11]

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

通过基于线性链条件随机场的中文分词的方法所使用的特征模板（如表2.3所示），以及本节所介绍的基于感知器的中文分词算法所使用的特征模板（如表2.4所示），可以看到基于感知器的方法可以使用词作为特征，而基于线性链条件随机场的方法只能使用字作为特征。因此在 2.3.3 节所介绍以字为单位作为分类目标的方法也称为基于字的中文分词算法，本节所介绍的以词为为基础的方法称为基于词的中文分词算法。

2.3.5 基于双向长短期记忆网络的中文分词

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而言，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**（Long Short-Term Memory, LSTM）[12, 13] 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。

LSTM 网络循环单元结构如图2.2所示，网络引入了新的内部状态（Internal State）$\boldsymbol{c}_t \in \mathcal{R}^D$，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** $\boldsymbol{f}_t$ 控制上一个时刻的内部状态 $\boldsymbol{c}_{t-1}$ 需要遗忘多少信息。**输入门** $\boldsymbol{i}_t$ 用来控制当前时刻的候选状态 $\tilde{\boldsymbol{c}}_t$ 有多少信息需要保存。**输出门** $\boldsymbol{o}_t$ 控制当前时刻内部状态 $\boldsymbol{c}_t$ 有多少信息需要输出给外部状态 $\boldsymbol{h}_t$。

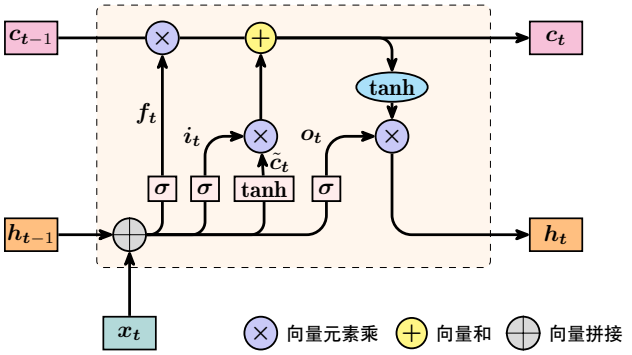

图 2.2　LSTM 网络循环单元结构

输入门 $\boldsymbol{i}_t$、输出门 $\boldsymbol{o}_t$ 和遗忘门 $\boldsymbol{f}_t$ 的计算方式为：

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i x_i + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \tag{2.5}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f x_i + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \tag{2.6}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o x_i + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \tag{2.7}$$

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

$$\tilde{\boldsymbol{c}}_t = \tanh(\boldsymbol{W}_c x_t + \boldsymbol{U}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c) \tag{2.8}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \otimes \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \otimes \tilde{\boldsymbol{c}}_t \tag{2.9}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \otimes \tanh(\boldsymbol{c}_t) \tag{2.10}$$

更为详细的介绍请参阅邱锡鹏教授《神经网络与深度学习》的第六章[14]。

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2.3给出了一个使用 BiLSTM 网络结合条件随机场（BiLSTM+CRF）进行分词的模型框架。

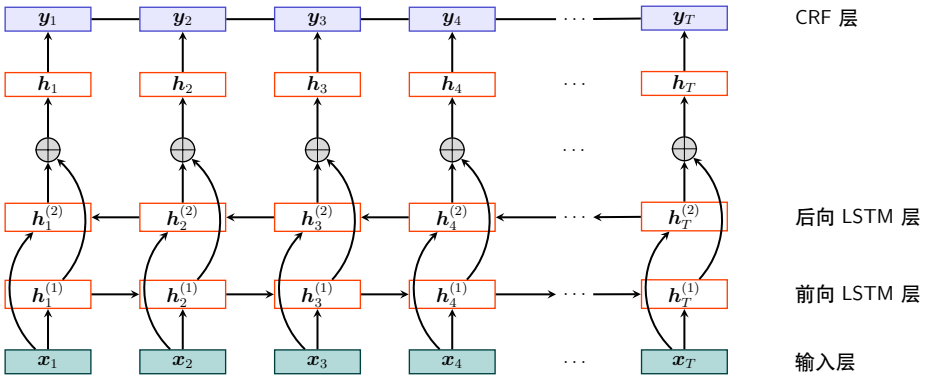

图 2.3 基于 BiLSTM+CRF 的神经网络分词模型框架

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 $x = \{c_1, c_2, \cdots, c_T\}$，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 $y = \{y_1, y_2, \cdots, y_T\}$。具体模型如图2.3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 $\boldsymbol{h}_1^{(1)}, \boldsymbol{h}_2^{(1)}, ..., \boldsymbol{h}_T^{(1)}$ 与反向 LSTM 隐状态序列的 $\boldsymbol{h}_1^{(2)}, \boldsymbol{h}_2^{(2)}, ..., \boldsymbol{h}_T^{(2)}$ 按位置进行拼接 $\boldsymbol{h}_i = \boldsymbol{h}_i^{(1)} \oplus \boldsymbol{h}_i^{(2)}$，从而得到完整的隐状态序列。

对于给定的长度为 T 的输入 $[\boldsymbol{x}]_1^T$，定义网络的输出矩阵为 $f_{\boldsymbol{\theta}}\left([x]_1^T\right)$（简写为 $f_{\boldsymbol{\theta}}$），其中 $[f_{\boldsymbol{\theta}}]_{i,t}$ 表示参数为 $\boldsymbol{\theta}$ 的网络对于句子 $[\boldsymbol{x}]_1^T$ 的第 t 个字的第 i 标签的打分。同时定义转移值矩阵 $\boldsymbol{A}$，其中 $[\boldsymbol{A}]_{i,j}$ 为相邻的两个字的标签从 i 标签到第 j 标签的值，$[\boldsymbol{A}]_{i,0}$ 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 $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} \cup \{[\boldsymbol{A}]_{i,j}, \forall i, j\}$。对于输入句 $[\boldsymbol{x}]_1^T$ 的某个特定标签序列 $[i]_1^T$ 定义得分为转移值和网络值的和，具体公式如下：

$$s\left([\boldsymbol{x}]_1^T, [i]_1^T, \tilde{\theta}\right) = \sum_{t=1}^{T}\left([\boldsymbol{A}]_{[i]_{t-1},[i]_t} + [f_\theta]_{[i]_t,t}\right) \tag{2.11}$$

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 $[j]_1^T$ 的得分进行归一化，得到标签序列的条件概率：

$$P\left([i]_i^T|[\boldsymbol{x}]_1^T,\tilde{\theta}\right)=\frac{e^{s\left([\boldsymbol{x}]_1^T,[i]_1^T,\tilde{\theta}\right)}}{\sum_{\forall[j]_1^T}e^{s\left([\boldsymbol{x}]_1^T,[j]_1^T,\tilde{\theta}\right)}} \tag{2.12}$$

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

$$\log P\left([y]_i^T|[\boldsymbol{x}]_1^T,\tilde{\theta}\right)=s\left([\boldsymbol{x}]_1^T,[y]_1^T,\tilde{\theta}\right)-\log\left(\sum_{\forall[j]_1^T}e^{s\left([\boldsymbol{x}]_1^T,[j]_1^T,\tilde{\theta}\right)}\right) \tag{2.13}$$

基于最大化对数似然目标，以及公式2.13的线性计算方法[15, 16]，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。

2.3.6 中文分词评价方法

中文分词算法效果评测通常也采用统计机器学习算法评测中常用的指标进行对比，包括：精确率（Precision，P）、召回率（Recall，R）、F 值（F-Measure）。各指标在中文分词任务中的具体计算方法如下：

$$\text{精确率（P）}=\frac{\text{算法输出的正确分词结果个数}}{\text{算法输出的全部分词结果个数}}\times 100\% \tag{2.14}$$

$$\text{召回率（R）}=\frac{\text{算法输出的正确分词结果个数}}{\text{测试集合中全部答案个数}}\times 100\% \tag{2.15}$$

$$\mathrm{F}_\beta=\frac{(\beta^2+1)\times P\times R}{P+R}\times 100\% \tag{2.16}$$

通常 F 值计算时设置 β 为 1，因此 F 值又称为 F1 值。

在 2.3.1 节中文分词概述中提到，未登录是中文分词任务的难点之一，也是影响中文分词算法效果的重要因素。因此，在中文分词评测中通常还会对召回率进一步细分为未登录词召回率（$\mathrm{R_{OOV}}$）和词典词召回率（$\mathrm{R_{IV}}$）。

$$\text{未登录词召回率（}\mathrm{R_{OOV}}\text{）}=\frac{\text{算法输出的未登录词正确结果个数}}{\text{测试集合中未登录词个数}}\times 100\% \tag{2.17}$$

未登录词召回率也是评价一个中文分词算法的主要指标

此外，由于有些中文分词算法会利用词典、无标注数据等除训练数据外的资源，为了能够更好的模型本身的效果进行评价，评测有时还会区分封闭测试（Closed Test）和开放测试（Open Test）。封闭测试仅允许使用给定的训练语料，而开放测试可以使用任意资源。

2.3.7 中文分词语料库

中文分词算法的训练通常依赖大规模标注语料。大规模中文分词语料集的建设也是推动中文分词算法快速发展的一个不可或缺的因素。SIGHAN 2005 和 SIGHAN 2008 是两组最常用的中文分词语料集合。SIGHAN 是国际计算语言学协会中文处理特别兴趣组，组织了多次包含多家机构的数据的中文处理相关评测（International Chinese Language Processing Bakeoff）。常见中文分词语料库如表2.5所示。本节将介绍目前较为广泛使用的部分中文分词语料集合。

表 2.5 常用中文分词语料库汇总

语料库名称	单词数量	简/繁体
北京大学分词语料库（PKU）	110 万	简体
香港城市大学分词语料库（CITYU）	145 万	繁体
微软研究院分词语料库（MSR）	237 万	简体
Academia Sinica（AS）	545 万	繁体
中文宾州树库 6.0（CTB 6.0）	78 万	简体
中文宾州树库 7.0（CTB 7.0）	120 万	简体
中文宾州树库 8.0（CTB 8.0）	162 万	简体
中文宾州树库 9.0（CTB 9.0）	208 万	简体
微博分词语料库（WordSeg-Weibo）	46 万	简体

1. 北京大学分词语料库（PKU）

北京大学分词语料库（也称为人民日报语料库）是由北京大学计算语言学研究所与富士通公司（Fujitsu）合作发布的包含 110 万字的分词语料集合。数据来源为《人民日报》，字符总数量约为 182 万。同时还制定了《现代汉语语料库加工规范》，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。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。

2. 香港城市大学分词语料库（CITYU）

香港城市大学分词语料库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。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。

3. 微软研究院分词语料库（MSR）

微软研究院分词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。该语料库将词汇分为三大类，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，包含日期，时间，持续时间，量词电话号码等。

2.4 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**（Part-of-speech Tagging，POS Tagging）是指在给定的语境中确定句子中各词的词性[17]。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。

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。这些具有多个词性的词语称为兼类词。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，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%，但是出现的次数缺却达到了 47%[18]。对 Brown 语料库的统计也发现超过 80% 的词通常只有一个词性。

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2.6列出了在宾州树库（Penn Treebank，PTB）中所使用的词性。而宾州大学汉语树库（Chinese Penn TreeBank，CPTB）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。

2.4.1 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[19]。

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：

表 2.6 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	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	专有名词单数
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

(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对训练语料进行分析，将错误的分析结果汇总为三元组 $<tag_a, tag_b, n>$ 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 n 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 $<tag_a, tag_b, n>$ 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。

2.4.2 基于隐马尔可夫模型的词性标注

隐马尔可夫模型（Hidden Markov Model，HMM）又称隐马尔科夫模型，是马尔可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。随着隐马尔可夫模型在语音识别领域取得巨大成功[20]，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。一个隐马尔可夫模型可用如下 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。$\pi = [\pi_i]_{1\times N}, \pi_i = P(q_1 = s_i), 1 \leqslant i \leqslant N$，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。$A = [a_{ij}]_{N\times N}, a_{ij} = P(q_{t+1} = s_j|q_t = s_i), 1 \leqslant i, j \leqslant N$，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。$B = [b_j(k)]_{N\times M}$，$b_j(k) = P(o_t = v_k|q_t = s_j), 1 \leqslant j \leqslant N, 1 \leqslant k \leqslant M$，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, 隐马尔可夫模型可以表示成 $\lambda = (A, B, \pi)$。M, N 也隐含的已经包含在 A, B, π 中。隐马尔可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 $O = \{o_1, o_2, ..., o_T\}$ 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[10]。

针对词性标注任务，使用隐马尔可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2.4给出了基于词性标注的隐马尔可夫模型概率图模型样例。

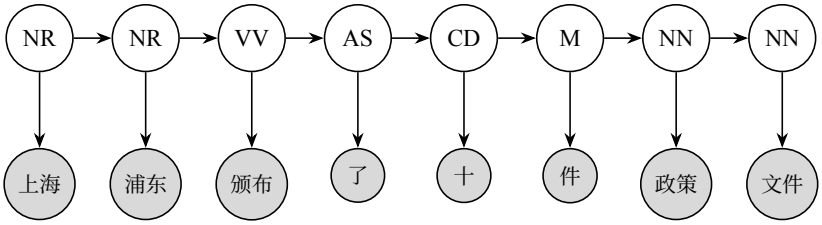

图 2.4 词性标注隐马尔可夫模型概率图模型样例

在实际应用过程中，使用隐马尔可夫模型进行词性标注，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。Gabriel García Márquez 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”，长度甚至超过 1000 个词。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类词和闭类词。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。

2.4.3 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[16] 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。基于卷积神经网络的词性标注算法神经网络结构如图2.5所示。

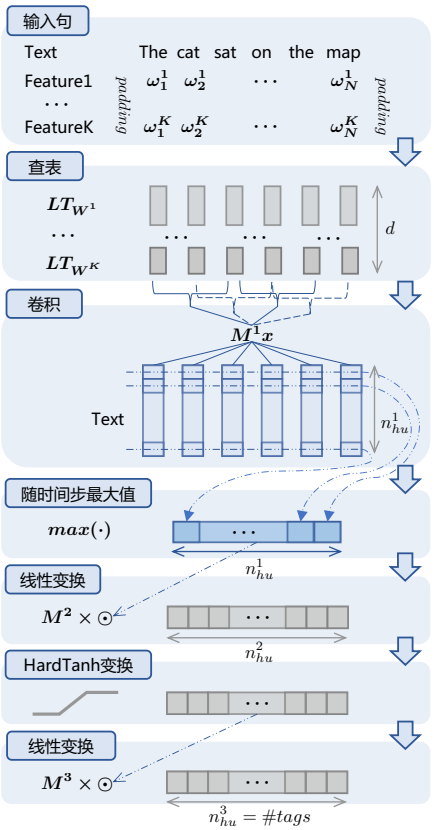

图 2.5　基于卷积神经网的词性标注模型结构

首先通过表查询（Lookup Table）$\mathrm{LT}_{\boldsymbol{W}}(\cdot)$ 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}。

$$\mathrm{LT}_{\boldsymbol{W}}(w) = \langle \boldsymbol{W} \rangle_w^1 \tag{2.18}$$

其中 $\boldsymbol{W} \in \mathbb{R}^{d_{wrd} \times |\mathbb{D}|}$，$\mathbb{D}$ 是包含有限个单词的的字典，$\langle \boldsymbol{W} \rangle_w^1 \in \mathbb{R}^{d_{wrd}}$ 表示 $\boldsymbol{W}$ 矩阵的第 w 列。$\boldsymbol{W}$

矩阵也是要进行学习的参数。对于给定的任意一句包含 T 个单词的句子 $[w]_1^t$，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：

$$\mathrm{LT}_{\boldsymbol{W}}([w]_1^T) = \left(\langle \boldsymbol{W}\rangle_{[w]_1}^1, \langle \boldsymbol{W}\rangle_{[w]_2}^1 ... \langle \boldsymbol{W}\rangle_{[w]_T}^1\right) \tag{2.19}$$

除了单词本身之外，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，可以将单词更一般得表示为 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。$\mathrm{LT}_{W^k}(\cdot)$ 是每维特征的查询表，$\boldsymbol{W}^k \in \mathcal{R}^{d_{wrd}^k \times |D^k|}$ 是第 k 维特征的嵌入向量矩阵，$d_{wrd}^k \in \mathcal{N}$ 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 $d_{wrd} = \sum_k d_{wrd}^k$，通过表查询得到连接后的向量：

$$\mathrm{LT}_{\boldsymbol{W}^1,...,\boldsymbol{W}^K}(w) = \begin{pmatrix} \mathrm{LT}_{\boldsymbol{W}^1}(w_1) \\ \vdots \\ \mathrm{LT}_{\boldsymbol{W}^K}(w_K) \end{pmatrix} = \begin{pmatrix} \langle \boldsymbol{W}^1\rangle_{w_1}^1 \\ \vdots \\ \langle \boldsymbol{W}^K\rangle_{w_K}^1 \end{pmatrix} \tag{2.20}$$

由此，可以得到如下表查询层输出矩阵：

$$\mathrm{LT}_{\boldsymbol{W}^1,...,\boldsymbol{W}^K}([w]_1^T) = \begin{pmatrix} \langle \boldsymbol{W}^1\rangle_{[w_1]_1}^1 & ... & \langle \boldsymbol{W}^1\rangle_{[w_1]_T}^1 \\ \vdots & & \vdots \\ \langle \boldsymbol{W}^K\rangle_{[w_K]_1}^1 & ... & \langle \boldsymbol{W}^K\rangle_{[w_K]_T}^1 \end{pmatrix} \tag{2.21}$$

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：

$$f_\theta^1 = \left\langle \mathrm{LT}_W([w]_1^T)\right\rangle_t^{d_{win}} = \begin{pmatrix} \langle \boldsymbol{W}\rangle_{[w]_{t-d_{win}/2}}^1 \\ \vdots \\ \langle \boldsymbol{W}\rangle_{[w]_t}^1 \\ \vdots \\ \langle \boldsymbol{W}\rangle_{[w]_{t+d_{win}/2}}^1 \end{pmatrix} \tag{2.22}$$

f_θ^1 会被送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：

$$\left\langle f_\theta^l\right\rangle_t^1 = \boldsymbol{W}^l \left\langle f_\theta^{l-1}\right\rangle_t^{d_{win}} + b^l \qquad \forall t \tag{2.23}$$

在同一层中 $\boldsymbol{W}^l$ 为相同参数。对于 f_θ^l 中每一维在公式2.23计算完成后，都要进行非线性变化，可

以采用如下方式：

$$\left[f_\theta^l\right]_i = \text{HardTanh}\left(\left[f_\theta^l\right]_i\right), \tag{2.24}$$

$$\text{HardTanh}(x) = \begin{cases} -1 & \text{if } x < -1 \\ x & \text{if } -1 \leqslant x \leqslant 1 \\ - & \text{if } x > 1 \end{cases} \tag{2.25}$$

通过公式2.23得到的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：

$$\left[f_\theta^l\right]_i = \max_x \left[f_\theta^{l-1}\right]_{i,t} \qquad 1 \leqslant i \leqslant n_{hu}^{l-1} \tag{2.26}$$

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2.24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:

$$f_\theta^l = \boldsymbol{W}^l f_\theta^{l-1} + b^l \tag{2.27}$$

在分类阶段，采用句子级别对数似然方法（Sentence-Level Log-Likelihood）。相关算法以及公式在第2.3.5节基于 BiLSTM-CRF 方法进行分词部分进行了详细介绍，也可通过参考文献 [16] 查看详细算法和公式推导。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。

2.4.4 词性标注评价方法

词性标注问题通常转换为多分类问题，并且每个单词仅会输出一个词性结果。因此，词性标注算法的评测通常采用准确率（Accuracy）和宏平均 F1（Macro-F1）两种评测指标。

准确率具体计算方法如下：

$$\text{准确率（Accuracy）} = \frac{\text{算法输出的正确结果个数}}{\text{算法输出的全部结果总数}} \times 100\% \tag{2.28}$$

宏平均 F1（Macro-F1）需要首先计算每个词性标签的 F1 值，再计算所有词性标签 F1 值的平

均值。以名词词性标签的精确率和召回率为例，具体计算方法如下：

$$P_{\text{名词}} = \frac{\text{算法输出的正确的名词结果个数}}{\text{算法输出的名词结果总数}} \times 100\% \tag{2.29}$$

$$R_{\text{名词}} = \frac{\text{算法输出的正确的名词结果个数}}{\text{测试集合中正确的名词结果总数}} \times 100\% \tag{2.30}$$

$$F1_{\text{名词}} = \frac{2 \times P_{\text{名词}} \times R_{\text{名词}}}{P_{\text{名词}} + R_{\text{名词}}} \tag{2.31}$$

宏平均 F1（Macro-F1）的具体计算方法如下：

$$\text{Macro-}F1 = \frac{1}{n} \sum_{i \in POS} F_i \tag{2.32}$$

其中 n 为词性标签的数量，POS 为词性标签集合。

从上述计算公式中可以看到，宏平均 F1 对含有较少单词的标签类别更敏感，每个类别标签的 F1 值同等重要。而准确率对于仅含有少量单词的标签类别的效果不敏感。从衡量算法的能力角度，宏平均 F1 相对可以更好的反映类别不均衡情况下的算法性能。

2.4.5 词性标注语料库

通过本章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。常见词性标注语料库如表2.7所示。本节将介绍几种常见的包含词性标签的语料库。

表 2.7 常见词性标注语料库汇总

语料库名称	单词数量	语言
英语宾州树库（PTB）	117 万	英文
通用依存树库（UD V2.0 CoNLL 2017）	281 万	多语言
RIT-Twitter	1 万	英文
ARK-Twitter	3 万	英文
中文宾州树库 6.0（CTB 6.0）	78 万	中文
中文宾州树库 7.0（CTB 7.0）	120 万	中文
中文宾州树库 8.0（CTB 8.0）	162 万	中文
中文宾州树库 9.0（CTB 9.0）	208 万	中文

1. 英语宾州树库

英语宾州树库（English Penn Treebank，PTB）是最知名和最常用的短语结构句法树库之一。在对句子的语法树标注的同时，也标注了句子中单词的词性信息。因此，英语宾州树库也是最常使用的词性语料库之一。该语料库中包含多个部分，其中 WSJ-PTB 是最常用于词性标注评测的集合，其原始数据来自于 1989 年的华尔街日报文章，按照 PTB(V2) 的标注策略进行标注，包含 49208 个句子，1173766 个单词，48 种不同的词性标签。

2. 中文宾州树库

中文宾州树库（Chinese Penn Treebank，CTB）是目前最常用的大规模中文短语结构句法标注语料库之一。1998 年开始构建，2016 年发布的最新的 Chinese Treebank 9.0 版本，包含中文新闻网站、政府文书、杂志文章、新闻群组、广播对话节目、博客等类不同来源的 3726 篇文章，共计 132076 个句子，2084387 个单词，3247331 个中文和外文字符。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。

3. 通用依存树库

通用依存树库（Universal Dependencies，UD）是一个为多种语言开发的跨语言一致的依存句法树库项目。其词性标注采用了 Google 通用词性标签[21]，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)。除了标记集之外，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。

2.5 延伸阅读

本章中介绍了中文分词和词性标注任务，这两个任务都是典型的序列标注任务，除了基于词典的中文分词算法之外，本章中介绍的其他算法都采用有监督分类算法。因此，这些方法通常都面临跨领域处理效果差、依赖大规模训练语料。此外，还存在基于特征表示的方法依赖人工设计特征函数，而经典深度学习模型无法有效利用知识等问题。针对上述问题，近年来有大量工作从不同方面开展研究。

针对中文分词任务，为了解决有监督分类算法依赖大规模训练数据的问题，大量的研究工作针对不同的分类算法开展了融合有标注数据和无标注数据的半监督算法研究，包括基于部分标签学习的条件随机场算法（Partial-label Learning with CRF）[22]、非参数贝叶斯模型（Nonparametric Bayesian）[23]、基于图的标签传播算法（Graph-based label propagation）[24]、协同训练方法（Co-training）[25] 等。为了解决深度学习方法不能有效利用已有知识的问题，近年来也有一些工作分别，从将字在词典中特征信息编码[26]、通过损失函数编码词典特征[27]、词典增强的自适应注意力

机制[28] 等方法将词典信息融合到深度学习模型中。针对中文分词问题中标注规范不统一的问题，多标注中文分词研究近年来也受到了越来越多的重视，研究者提出了对抗标注准学习（Adversarial Multi-Criteria Learning ）[29]、转换长短时记忆网络（Switch-LSTMs）[30] 等方法试图同时学习多种分词标准。为了使得深度学习模型可以更好的融合词语级别特征，研究者提出了基于转移中文分词模型[31]、栅格化循环神经网络[32] 等方法。针对如何更好的评价中文分词算法在不同情况下的性能以及不同场景下的鲁棒性问题，一些工作从细粒度的中文分词算法评价[33] 以及中文分词算法鲁棒性研究[34] 等方面开展研究。

针对词性标注任务，为了能够利用大规模的无标注数据，先后提出了期望正规化（Expectation Regularization）[35]、基于平均感知器半监督算法[36]、半监督密集近邻（Semi-supervised condensed nearest neighbor）[37]、基于图信号的半监督主动学习（Active semi-supervised learning）[38] 等方法。针对算法训练数据和应用领域不同时，性能大幅度降低的问题，跨领域和领域自适应方法在词性标注任务中也有大量的研究工作，包括基于鲁棒表示的方法[39]、基于熵的数据选择方法[40]、分层贝叶斯（Hierarchical Bayesian）[41]、基于对抗神经网络算法[42]、基于强化学习的训练数据选择方法[43] 等。此外，针对汉语的词性标注标注问题，通常采用流水线方式，首先对汉语句子进行分词，之后在对分词结果进行词性标注。由于，流水线模式会造成错误的传递问题，因此也有一些工作将中文分词和词性标注问题联合建模，包括分类目标合并[44]、级联线性模型（Cascaded linear model ）[45]、词格重排序（Word lattice reranking ）算法[46]、堆叠子词模型[47]、双向注意力机制[48] 等方法。

中文分词和词性标注是自然处理的底层任务，其分析效果对后续任务有很大影响，长期以来都是自然语言处理研究的重点。此外，中文分词和词性标注也是典型的序列标注任务，结构化机器学习领域很多工作也将上述任务作为算法效果验证目标。

2.6 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) 英语中句子切分主要解决什么歧义问题？如何使用有监督分类算法进行句子切分？
(3) 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。
(4) 如何在基于线性链条件随机场的中文分词算法中引入词典特征？
(5) 如何处理词性标注算法中的未登录词？
(6) 如何同时 BiLSTM-CRF 方法进行分词和词性标注联合建模？

参考文献

[1] Fromkin V, Rodman R, Hyams N. An introduction to language[M]. Cengage Learning, 2018.

[2] Francis W N. A tagged corpus–problems and prospects[J]. Studies in English linguistics for Randolph Quirk, 1980:192-209.

[3] Jurafsky D, Martin J H. Speech and language processing: An introduction to speechrecognition, natural language processing and computational linguistics[M]. 2nd ed. Pearson, 2008.

[4] Porter M F. An algorithm for suffix stripping[J]. Program, 1980.

[5] 俞士汶, 段慧明, 朱学锋, 等. 北大语料库加工规范：切分·词性标注·注音[M]. 北京大学计算语言学研究所, 2003.

[6] 黄昌宁, 李玉梅, 朱晓丹. 中文文本标注规范 (5.0 版)[M]. 微软亚洲研究院, 2006.

[7] 刘开瑛. 中文文本自动分词和标注[M]. 商务印书馆, 2000.

[8] 黄昌宁, 赵海. 中文分词十年回顾[J]. 中文信息学报, 2007, 21(3):8-19.

[9] 宗成庆. 统计自然语言处理[M]. 北京: 清华大学出版社, 2013.

[10] 李航. 统计学习方法（第二版）[M]. 清华大学出版社, 2019.

[11] Zhang Y, Clark S. Chinese segmentation with a word-based perceptron algorithm[C]//Proceedings of the 45th annual meeting of the association of computational linguistics. 2007: 840-847.

[12] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8):1735-1780.

[13] Gers F A, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with lstm[J]. Neural computation, 2000, 12(10):2451-2471.

[14] 邱锡鹏. 神经网络与深度学习[M/OL]. 北京: 机械工业出版社, 2020. https://nndl.github.io/.

[15] Rabiner L R. A tutorial on hidden markov models and selected applications in speech recognition[J]. Proceedings of the IEEE, 1989, 77(2):257-286.

[16] Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch[J]. Journal of machine learning research, 2011, 12(ARTICLE):2493-2537.

[17] 吴立德. 大规模中文文本处理[M]. 复旦大学出版社, 1997.

[18] 张虎, 郑家恒, 刘江. 语料库词性标注一致性检查方法研究[J]. 中文信息学报, 2004, 18(5):12-17.

[19] Brill E. A simple rule-based part of speech tagger[C/OL]//ANLC '92: Proceedings of the Third Conference on Applied Natural Language Processing. USA: Association for Computational Linguistics, 1992: 152–155. https://doi.org/10.3115/974499.974526.

[20] Bahl L, Brown P, De Souza P, et al. Maximum mutual information estimation of hidden markov model parameters for speech recognition[C]//ICASSP'86. IEEE International Conference on Acoustics, Speech, and Signal Processing: volume 11. IEEE, 1986: 49-52.

[21] Petrov S, Das D, McDonald R. A universal part-of-speech tagset[C]//Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12). 2012: 2089-2096.

[22] Yang F, Vozila P. Semi-supervised chinese word segmentation using partial-label learning with conditional random fields[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014: 90-98.

[23] Fujii R, Domoto R, Mochihashi D. Nonparametric bayesian semi-supervised word segmentation[J]. Transactions of the Association for Computational Linguistics, 2017, 5:179-189.

[24] Zeng X, Wong D F, Chao L S, et al. Graph-based semi-supervised model for joint chinese word segmentation and part-of-speech tagging[C]//Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2013: 770-779.

[25] Zhang L, Wang H, Sun X, et al. Exploring representations from unlabeled data with co-training for chinese word segmentation[C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013: 311-321.

[26] Zhang Q, Liu X, Fu J. Neural networks incorporating dictionaries for chinese word segmentation [C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 32. 2018.

[27] Liu J, Wu F, Wu C, et al. Neural chinese word segmentation with lexicon and unlabeled data via posterior regularization[C]//The World Wide Web Conference. 2019: 3013-3019.

[28] Zhao X, Yang M, Qu Q, et al. Improving neural chinese word segmentation with lexicon-enhanced adaptive attention[C]//Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020: 1953-1956.

[29] Chen X, Shi Z, Qiu X, et al. Adversarial multi-criteria learning for chinese word segmentation[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017: 1193-1203.

[30] Gong J, Chen X, Gui T, et al. Switch-lstms for multi-criteria chinese word segmentation[C]// Proceedings of the AAAI Conference on Artificial Intelligence: volume 33. 2019: 6457-6464.

[31] Zhang M, Zhang Y, Fu G. Transition-based neural word segmentation[C/OL]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, 2016: 421-431. https://www.aclweb.org/anthology/P16-1040. DOI: 10.18653/v1/P16-1040.

[32] Yang J, Zhang Y, Liang S. Subword encoding in lattice lstm for chinese word segmentation[C]// 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: 2720-2725.

[33] Fu J, Liu P, Zhang Q, et al. Rethinkcws: Is chinese word segmentation a solved task?[C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020: 5676-5686.

[34] Wang X, Liu Q, Gui T, et al. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing[C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations. 2021: 347-355.

[35] Mann G S, McCallum A. Simple, robust, scalable semi-supervised learning via expectation regularization[C]//Proceedings of the 24th international conference on Machine learning. 2007: 593-600.

[36] Hajic J, Raab J, Spousta M, et al. Semi-supervised training for the averaged perceptron pos tagger [C]//Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009). 2009: 763-771.

[37] Sogaard A. Semi-supervised condensed nearest neighbor for part-of-speech tagging[C]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011: 48-52.

[38] Gadde A, Anis A, Ortega A. Active semi-supervised learning using sampling theory for graph signals [C]//Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014: 492-501.

[39] Schnabel T, Schütze H. Flors: Fast and simple domain adaptation for part-of-speech tagging[J]. Transactions of the Association for Computational Linguistics, 2014, 2:15-26.

[40] Song Y, Klassen P, Xia F, et al. Entropy-based training data selection for domain adaptation[C]// Proceedings of COLING 2012: Posters. 2012: 1191-1200.

[41] Finkel J R, Manning C D. Hierarchical bayesian domain adaptation[C]//Proceedings of human language technologies: The 2009 annual conference of the North American Chapter of the Association for Computational Linguistics. 2009: 602-610.

[42] Gui T, Zhang Q, Huang H, et al. Part-of-speech tagging for twitter with adversarial neural networks [C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017: 2411-2420.

[43] Liu M, Song Y, Zou H, et al. Reinforced training data selection for domain adaptation[C]// Proceedings of the 57th annual meeting of the association for computational linguistics. 2019: 1957-1968.

[44] Ng H T, Low J K. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based?[C]//Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. 2004: 277-284.

[45] Jiang W, Huang L, Liu Q, et al. A cascaded linear model for joint chinese word segmentation and part-of-speech tagging[C]//Proceedings of ACL-08: HLT. 2008: 897-904.

[46] Jiang W, Mi H, Liu Q. Word lattice reranking for chinese word segmentation and part-of-speech tagging[C]//Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008). 2008: 385-392.

[47] Sun W. A stacked sub-word model for joint chinese word segmentation and part-of-speech tagging [C]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011: 1385-1394.

[48] Tian Y, Song Y, Ao X, et al. Joint chinese word segmentation and part-of-speech tagging via two-way attentions of auto-analyzed knowledge[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 8286-8296.

索引