自然语言处理导论

张奇 桂韬 黄萱菁

2023 年 1 月 6 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

4. 语义分析

掌握一种语言意味着懂得如何产生并理解数量无限的该种语言句子的意义。研究语言意义的科学被称为**语义学**（Semantics）。语义问题也被大多数语言学家认为是语言的核心问题。同时也受到了包括哲学、逻辑学、心理学以及计算机等众多学科广泛关注。自然语言处理目标就是要使计算机具有理解和运用自然语言的能力。因此，语义也是自然语言处理的关键问题和难点问题。从计算角度，语义研究需要以对语义的形式化结构表示为基础。这种形式化结构表示称之为**语义表示**（Semantic Representation）。自然语言处理中**语义分析**（Semantic Analysis）则是指解释各粒度的语言单位，并将其转换为对应语义表示。

本章首先介绍语义学和语义表示的基本概念和主要研究内容，在此基础上语义和知识的表示方法，词义消歧算法，以及语义角色标算法。

4.1 语义学概述

什么是“意义”是一个困扰了哲学家和语言学家数千年的问题。我们可以非常容易地理解中文，并且用汉字组成对其他人来说也是有意义的句子。我们也可以知道某个词语、句子是否有意义，还可以通过一个句子衍推出另外一个句子。意义从何而来？语言的意义的本质又是什么？学术界对这些问题众说纷纭没有定论。中国古代以“字”为核心的训诂语义研究达到了很高的水准，公元前2世纪就有了专门解释词义的专著《尔雅》。先秦时期，荀子和墨子就开始对“名”与“实”的关系进行讨论。古希腊哲学家苏格拉底、亚里士多德等也都在其哲学著作中探讨过语言的意义。

语义学的研究目标就是发现和阐述关于意义的知识。1883年由法国语言学家布雷亚尔（Michel Bréal）发表的论文中首次提出了语义学的概念，1897年出版了《语义学探索》对语义学的研究对象和可能采取研究方法进行了系统地阐述，从此语义学逐渐成为语言学中一门独立的学科。语义学的研究已经成为语言学、逻辑学、哲学、心理学、认知科学、人工智能等多门学科的研究热点和难点。也因此语义学研究十分庞杂，理论和流派层出不穷，不同学科关于语义学的研究范畴、关注角度、重点问题都有很大差异。本章侧重从语言学和自然语言处理角度，对语义问题的基本概念和任务进行介绍。

从语言表达层面划分，语义学的研究大致可以分为三个层面：（1）**词汇语义学**（Lexical Semantics）

，主要包括词义问题、词汇间关系、词汇场、成语的语义等；（2）**句子语义学**（Sentential Semantics），主要以真值条件语义理论、配价理论、生成理论等为基础研究句义关系以及语序等问题；（3）**话语语义学**（Discourse Semantics），主要研究句子以上层次结构的意义，包括话语衔接、话语的连贯、语用过程解释等。本节中针对自然语言处理领域关注较多的词汇语义学和句子语义学基本语言学理论进行介绍。

4.1.1 词汇语义学

词是语言中能够能够独立运用的最小的单位，也是音、形、义的结合体。词语通过搭配组合，可以构建出短语、句子、篇章等复杂的语言的结构。语义学自创建之初，就将词汇语义作为重要的研究目标。**词汇语义学**主要研究单个词语的意义以及词汇之间的相互关系。

1. 词汇语义理论

词义（Word Meaning）有很多的方面，可以用从不同的角度分析和定义，因而出现了包括语义场理论、语义成分分析、并置理论、框架语义理论等众多词汇语义理论。简要对其中典型的词汇语义理论进行介绍。

语义场理论（Semantic Field）也称作词义场理论（Lexical Field）认为语言中词汇的意义是相互联系的，构成一个完整的系统和网络，具有某些相同语义特征的一组词聚而成场。如表示苹果、香蕉、橘子等都有一个共同的义素 [+ 水果]，组成了语义场中的“水果场”（Fruit Field ）。水果、肉、蔬菜、谷物等又可以构成食物语义场，如表4.1所示。根据语义场理论，不能够孤立的研究一个词的词义，只有通过分析比较词于词之间的关系，才能确定一个词的真正意义。除了词汇的上下位关系外，同义关系、反义关系等都构成语义场。因此，语义场也可以认为是研究词与词之间的聚合关系（Paradigmatic Relation）。

表 4.1　食物语义场示例

食物	水果	苹果
		香蕉
		...
	肉	牛肉
		羊肉
		...
	蔬菜	白菜
		菠菜
		...
	谷物	大米
		小麦
		...

语义成分分析（Componential Analysis）理论认为词义可以由最小的语义成分组合而成。这种最小的语义成分又被成为语义特征。例如，可以定义 ADULT、YOUNG、MALE、FEMALE 为语义特征，根据这些特征可以表达词汇的意义：

man: ADULT + MALE

woman: ADULT +FEMALE

boy: YOUNG + MALE

girl: YOUNG +FEMALE

词的语义特征可以从语法-语义特征、内在语义特征以及感受性语义特征等三个部分进行考察。语法-语义特征主要指明语法标注，如人称、性、数、语态等；内在语义特征指直接反映的客观事务本质的语义特征；感受性语义特征指带有主观色彩和表示内涵的语义特征[1]。

义元理论（Theory of Lexcial Primitives ）的核心思想是自然语言中包含非常少部分的词语，这些词语可以用于解释绝大部分词汇的意义。这些语义上不能分解的最小的意义组成单元称为**义元**（Lexcial Primitives）。例如：man 和 fish 是义元，而 fishy 和 manliness 则是衍生词。可以使用义元对其他词语进行解释[2]，例如：

boy: young human being that one thinks of as becoming a man.

girl: young human being that one thinks of as becoming a woman.

woman: human being that could be someone's mother.

man: human being that could casuse a woman to be someone's mother.

董振东教授所创建的知网（HowNet）[3] 也结合了义元理论，构建了包含 2540 多个义元的精细的语义描述体系，并为 237974 个汉语和英语词所代表的概念进行了标注。例如：

美味: edible| 食物:modifier=GoodTaste| 好吃

难题: problem| 问题:modifier=difficult| 难

HowNet 中义元采用中英双语的形式进行描述，上述例子中"edible| 食物"、"GoodTaste| 好吃"义元。"难题"是由核心义元"problem| 问题"以及对核心义元的附加描述义元"difficult| 难"组成。

框架语义学（Frame Semantics）则认为词义只能在相应的知识框架背景中才能得到理解。在意义的理解过程中，概念并不是杂乱无章的，很多概念往往具有一种同现的趋势，例如：顾客、服务员、吃饭、账单等概念都与饭店相关，是理解"饭店"的框架。此外，在意义的理解过程中，并不一定需要激活一个语义框架的全部成分，往往只需要激活部分框架。例如：在 Fillmore 和 Atkins 所给出的 RISK 的框架[4] 是由如下成分组成：

RISK frame:

- Chance (uncentainty about the future)
- Harm
- Victim (of the harm)
- Valued Object (potentially endangered by the risk)

Situation (which gives rise to the risk)
Deed (that brings about the Situation)
Actor (of the Deed)
Gain (by the Actor in taking the risk)
Purpose (of the Actor in the Deed)
Beneficiary and motivation (for the Actor)

对于包含动词 risk 的句子，可以根据 RISK 框架对其进行理解。例如：

(1) You've (Actor/Victim) risked your health (Vauled Object) for a few cheap thrills (Gain).

(2) She (Actor/Victim) has risked so much (Valued Object) for the sake of vanity (Motivation).

框架网络（FrameNet）[5] 就是根据框架语义学理论，依靠语料库的支持构建的词汇语义知识库。截止 2022 年 3 月，FrameNet 针对 13685 个词元（lexical unit）构建了 1224 语义框架。

2. 词汇间的关系

词汇之间的关系（Lexcial Relations）是词汇语义学研究的另一个重点问题。关系类型可以分为三大类：形体关系（Form Relations）、意义关系（Sense Relations）和实体关系（Object Relations）[6]。**形体关系**主要研究词汇的声音形体和拼写之间的关系。**意义关系**主要关注词汇意义之间的关联性、相似性、对立性等关系。**实体关系**则主要研究词汇之间的客观关系。表4.2给出了词汇之关系的主要类型和例子。

表 4.2　词汇之间的主要关系

词汇之间的关系类型			示例
形体关系	Homonymy	同音异义关系	ring 与 wring 发音相同
		同形异义关系	bank（银行）与 bank（河岸）词形相同
意义关系	Synonymy	同义关系	“电脑”与“计算机”是同义词
	Antonymy	反义关系	“大”与“小”之间是反义关系
	Hyponymy	下位关系	“燕子”是“鸟”的下位词
	Hypernymy	上位关系	“动物”是“老虎”的上位词
实体关系	Meronymy	部分整体关系	“发动机”与“汽车”

同音异义关系和同形异义关系不涉及词汇的意义，表示具有相同发音或者相同形式但是意义不同的词汇。同义关系表示两个词汇含有相同或相近的意义（即义项）。通常认为两个词只有一个义项相同，就可以被认为是同义词。反义关系表示意义相对立或相反的词。上位关系和下位关系表示词的意义包含另外一个词，或者词的意义包含在另外一个词中。许多下位词可以属于同一个上位词。部分整体关系表示客观实体之间的组成部分和整体之间的关系。同形异义关系和同义关系也反映了词汇的形式（word form）和义项（sense）之间的分离。

根据词汇间关系的研究，美国普林斯顿大学 George A. Miller 教授领导构建了 WordNet[7]，是

目前最常用的英语词汇知识资源库。在其中词汇按照义项组合成同义集（Synset），每个义项表达不同的概念。名词、动词、形容词和副词各自独立的组合成网络。WordNet 的语义关系不是在单词之间建立的，而是在义项之间建立的。义项之间的关系包括：同义关系 (synonymy)、反义关系 (antonymy)、上下位关系 (hypernymy/hyponymy)、整体与部分关系 (meronymy/holonymy)、对等关系（coordinate terms）、继承关系 (entailment) 等类型。WordNet 3.1 版本包含 155327 个单词，175979 个同义集，共组成了 207016 对单词和义项对。

“bank”做为名词和动词在 WordNet 中的部分词条如下：

Noun

1. bank (sloping land (especially the slope beside a body of water)) “they pulled the canoe up on the bank”; “he sat on the bank of the river and watched the currents”
2. depository financial institution, bank , banking concern, banking company (a financial institution that accepts deposits and channels the money into lending activities) “he cashed a check at the bank”; “that bank holds the mortgage on my home”
3. bank (a long ridge or pile) “a huge bank of earth”
4. bank, cant, camber (a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force)

Verb

1. bank (tip laterally) ”the pilot had to bank the aircraft”
2. bank (enclose with a bank) ”bank roads”
3. bank (do business with a bank or keep an account at a bank) ”Where do you bank in this town?”
4. deposit, bank (put into a bank account) ”She deposits her paycheck every month”

在 WordNet 中单词的每个义项都包含采用字典风格的注解和该义项的同义集，部分义项还包含用例。例如：bank^4（bank 的第 4 个义项）与 cant^4 和 camber^2 组成了一个同义集，对这个同义集的描述是 a slope in the turn of a road or track。

此外，在 WordNet 中名词和动词可以根据上下位关系或者部分整体关系构成层级结构。例如，bank^4 的上下位关系链：

bank, cant, camber (a slope in the turn of a road or track)
=> slope, incline, side
=> geological formation, formation
=> object, physical object
=> physical entity
=> entity

在后续的发展中，WordNet 逐渐发展为支持 200 多种语言的语言知识库。

4.1.2 句子语义学

句子语义学主要是在句子层面对意义的研究。人们通常通过句子来表达完整语义，相较于词汇句子也复杂的多，因此非常多的工作都是围绕句子语义学从各个角度开展，包括语音、语法、逻辑、认知、心理学等等。本节中，从语言学角度对句子语义学的主要理论进行简要介绍。

1. 句子语义理论

语言是对外部世界的编码，句子就是人们对客观世界的概念表征，人们对句子意义的认知始于真假判断。真值条件语义学（Truth-conditional Semantics）核心就是将意义定义为一个句子或句子所表达的命题为真时所必须满足的一系列条件[8]。该理论试图通过解释句子何时为真来定义给定句子或命题的意义。提出了一个检验句子真值的通用公式—T 公式：S is true iff P，S 代表某个句子。P 代表句子的真值条件，iff 表示“if and only if”。例如：他是学生，S 表示这个句子，P 表示“他”所代表的人并且真的是学生的列表。真知条件语义学开创了用数理逻辑方法解释自然语言的语义，用严格数学方法研究自然语言语义的方向。但自然语言中很多句子并不能判断真假，如疑问句和所使句等，这也在一定程度上限制了真知条件语义学的应用。

在词汇语义理论中语义成分分析理论认为词义可以由最小的语义成分组合而成，在句子层面同样也存在“语义成分”。这种语义成分通常称作“语义格”（Semantic Case）。格语法（Case Grammar）以及从格语法发展而来的框架语义学（Frame Semantics）都是以语义格为基础。语义格也称语义角色（Semantic roles），又称语义关系、主题关系（Thematic relations）。美国语言学家菲尔墨（Charles J. Fillmore）对乔姆斯基的转换语法进行了延伸，提出了格语法，认为句子中名词短语总是与动词相关，并且以唯一可以识别的方式表示了名词短语的语义格。他指出“主语”、“宾语”等语法关系实际上都是表层结构上的概念，语言的底层是用“施事”、“受事”、“工具”等概念所表示的句法语义关系。

例如：The key opened the door.

The boy opened the door with a key.

上述例子中的 key 在深层句法语义上始终是“工具”，但是它可以是主语，也可以是介词 with 的宾语。常见的语义格如表4.3所示。语义格的数量以及定义并没有定论，甚至在菲尔墨的不同文章中也有不同。

在格语法中对于词库中词汇的每个词条需要标明其格特征，对于名词标明其可以作为的语义格（例如：“街道”需要标明 [+LOCATION]），对于动词需要标明其对应的格框架。格框架由主要概念和辅助概念组成，主要概念通常为动词，辅助概念为施事格、受事格、方位格、工具格等语义深层格。

例如：BREAK 可以放入如下格框架：

[(施事格) (受事格) (工具格) (方位格)]

格框架可以帮助解决之前语法中隐藏的某些歧义，可以提取意义相同但是结构不同的句子。

表 4.3 常见语义格定义

名称	定义	示例
施事格 AGENT	动作的发起者	**He** wrote the book.
受事格 PATIENT	受到动作或状态影响的实体	He wrote **the book**.
工具格 INSTRUMENT	对动作或状态而言作为某种因素而牵涉到的无生命的客体	He cleaned the table with **an antiseptic wipe**.
方位格 LOCATION	动作或状态的处所或空间位置	They are in the **building**
来源格 SOURCE	动作所作用到的事物的来源或发生位置变化过程中的起始位置	He bought a book from **Jeremy**.
目标格 GOAL	动作所作用到的事物的终点或发生位置变化过程中的终端位置	He sold a book to **Jeremy**.
时间格 TIME	动作或状态的时间	**In the winter**, he saw snow.
伴随格 COMITATIVE	与施事共同完成动作的伴随者	He sang a song with **Jeremy**.
受益者格 BENEFICIARY	因动作的实行而受益的实体	He sang a song for **Jeremy**.
感受格 EXPERIENCER	知道谓语所描述的动作或状态，但不受该动作或状态控制	**He** saw snow.

例如：他在房间里用锤子打破了玻璃杯。

根据 BREAK 框架得到：

[BREAK [Case-frame:

[AGENT: 他

PATIENT: 玻璃杯

INSTRUMENT: 锤子

LOCATION: 房间]]]

2. 句义关系

句子之间也存在各种语义关系，把句子当做一个整体，句子和句子之间的语义关系可以包含同义、反义、蕴含等。

同义关系（Synonym）表示两个不同的句子表达相同的意义。

例如：a. 他打碎了玻璃杯。

b. 玻璃杯被他打碎了。

上述两个句子表达了相同的含义，具有相同的真值。

反义关系（Inconsistency）表示两个句子的意义只能有一个与客观事实相符。

例如：a. 他打碎了玻璃杯。

b. 玻璃杯完好的放在橱窗里。

上述两个句子表达的含义，一句为真时，另外一句一定为假。

蕴含关系（Entailment）表示两个句子的意义，前者为真时后者必然为真，前者为假时后者可能为真也可能为假。

例如：a. 他拿着一本书去了校门口。

b. 书在他手里。

上述例子中，句子 a 为真，那么句子 b 也必然为真，我们可以说 a 蕴含 b。

预设关系（Presupposition）表示一个句子的意义是另外一个句子的前提。

例如：a. 复旦大学江湾校区管委会举办了迎新活动。

b. 复旦大学有多个校区。

上述例子中，句子 a 为真，那么句子 b 也必然为真，如何 a 句为假，b 句仍然为真。预设关系通常认为是语用关系，两者有前后的时间关系，属于历时关系。

4.2 语义表示

语义表示（Semantic Representation）是语义的符号化和形式化的过程，主要研究语义表示的通用原则和方法。为了使得计算机能够处理自然语言的语义，就需要用恰当的模式对语义进行表示，因此语义表示方法也是自然语言理解的基础。从上一节中所介绍的语义学基础理论，可以看到目前关于意义的定义和本质还没有定论，大量的语义学理论从不同的角度开展了一系列的讨论。已有的语义表示方法大多都是根据不同的语义学理论针对某项具体研究时所提出，有一定的针对性和局限性，适用于词汇、句子、篇章等各个层面各种应用的通用语义表示方法还是一个亟待解决的问题。本节中介绍常见的一阶谓词逻辑、框架、语义网等语义表示方法，分布式表示表示方法在下节中单独介绍。

4.2.1 谓词逻辑表示法

数理逻辑（Mathematical Logic）在知识的形式化表示和机器的自动定理证明方面都有广泛的应用和很好的表现，真值条件语言学中也是使用用数理逻辑来研究自然语言的语义。自然语言的语义表示中也经常数理逻辑的方法。其中常用的是谓词逻辑（Predicate Logic）和命题逻辑（Propositional Logic）。谓词逻辑可以更细致的刻画语义，可以表示事务的状态、属性、概念等事务性语义，也可以表示因果关系等规则性语义，同时命题逻辑也可以认为是谓词逻辑的一种特例，因此本节中重

点介绍谓词逻辑。

在谓词逻辑中，研究对象全体所构成的非空集合称为论域（个体域）。论域中的元素称为个体或个体词。论域中包含的个体数量可以是无限的也可以是有限的。个体可以是常量、变量或函数。个体常量表示具体的或特定的个体；个体变量表示抽象的或泛指的个体；个体函数表示一个个体到另一个个体的映射。用于刻画个体的性质、状态或个体之间关系的词项则称为谓词。这些常量、变量、函数和谓词也都需要有明确的语义解释。

谓词一般用 $P(x_1, x_2, \cdots, x_n)$ 表示，P 是谓词名，$x_1, x_2, \cdots, x_n$ 表示某个独立存在的事务或某个抽象的概念。如果谓词 P 中的所有个体都是常量、变量或函数，则称该谓词为一阶谓词（First Order Predicate Logic）。如果某个个体本身又是一个一阶谓词，则称 P 为二阶谓词。例如：

谓词：Teacher(x) 表示 x 是教师，是一阶谓词。

句子：“老张是一名老师”可以表示为 Teacher(老张)

除了直接使用单个谓词和指代对象的常量、变量或者函数组成原子公式之外，还可以使用 5 种逻辑连接词和量词构造复杂的表示，就是谓词逻辑中的公式。原子公式是谓词演算的基本组块，运用连接词可以组合多个原子公式，以构成更加复杂的公式。具体连接词和量词定义如下：

(1) 连接词

$\neg$：“否定”（Negation）或“非”

$\vee$：“析取”（Disjunction）或“或”

$\wedge$：“合取”（Conjunction）或“与”

$\rightarrow$：“蕴含”（Implication）或“条件”

$\leftrightarrow$：“等价”（Equivalence）或“双向蕴含”

连接词的真值表如表4.3所示。连接词的优先级从高到底排列为：$\neg$、$\wedge$、$\vee$、$\rightarrow$、$\leftrightarrow$。

(2) 量词

$\forall$：全称量词（Universal Quantifier），表示对个体域中的所有（或任意一个）个体 x

$\exists$：存在量词（Existential Quantifier），表示在个体域中存在个体 x

表 4.4　连接词真值表

P	**Q**	$\neg P$	$P \vee Q$	$P \wedge Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
T	T	F	T	T	T	T
T	F	F	T	F	F	F
F	T	T	T	F	T	F
F	F	T	F	F	T	T

可以利用上述谓词和逻辑公式的定义对如下句子的语义进行符号化表示：

(1) “有机器人都是红色的”

谓词定义：ROBOT(X) 表示 X 是机器人；COLOR(X,Y) 表示 X 的颜色为 Y

谓词公式：$(\exists X)[ROBOT(X) \wedge COLOR(X, RED)]$

(2) “人人都爱护环境”

谓词定义：MAN(X) 表示 X 人；PROTECT(X,Y) 表示 X 保护 Y

谓词公式：$(\forall X)[MAN(X) \rightarrow PROTECT(X, ENVIRONMENT)]$

(3) “小明不在 3 号房间”

谓词定义：INROOM(X,Y) 表示 X 在 Y 中

谓词公式：$\neg INROOM(XIAOMING, ROOM3)$

由于谓词逻辑具有扎实的数学基础，一阶谓词逻辑具有充分的表达能力和完备的逻辑推理算法，其推理过程和结果的准确性可以得到有效保证，因此可以精密地表达语义，有很广泛的应用领域。但是使用一阶谓词逻辑表示语义并不简单，通常需要如下步骤：

(1) 定义谓词及个体：确定每个谓词及个体的确切含义。

(2) 变量赋值：根据所要表达的事务或概念，为每个谓词中的变量赋予特定的值。

(3) 谓词公式构造：根据所有表达的语义，用适当的连接符号和量词将各谓词连接起来。

可以看到如果使用一阶谓词逻辑清晰表达语义，需要对大量谓词以及个体，不仅涉及到领域内特定知识的定义，还涉及到领域的通用知识甚至是世界知识的定义，过程十分庞杂并且需要领域专家的协助。此外，一阶谓词逻辑不能很好的表达非精确的语义，以及在推理过程中很可能产生的组合保证问题也都限制了其应用范围。

4.2.2 框架表示法

框架（Frame）表示法是以框架语义理论为基础发展起来的一种语义表示方法。框架用来表示所讨论对象（一个事务、概念或者事件）的语义。每个框架由若干“槽”（Slot）组成，描述框架所讨论对象的某一方面的属性。每个槽根据实际情况可以赋值一定类型的实例或若干数据，称为槽值。每个槽还可以划分为若干“侧面”（Facet），描述相应属性的一个方面。每个侧面也可以赋值一定类型的实例或若干数据，称为为侧面值。一个框架通常包含多个不同的槽和侧面，分别用框架名、槽名和侧面名表示。典型的框架结构如下所示：

< 框架名 >

< 槽名 1>

< 侧面名 1-1>< 值 1-1> ...

< 侧面名 1-2>< 值 1-2> ...

...

< 槽名 2>

< 侧面名 2-1>< 值 2-1> ...

< 侧面名 2-2>< 值 2-2> ...

...

...
< 槽名 n>
< 侧面名 n-1> < 值 n-1> ...
< 侧面名 n-2> < 值 n-2> ...
...

MUC-3 事件语义理解的评测集合上所定义的“恐怖袭击事件”框架[9] 如下所示：

框架名：恐怖袭击事件
槽 1：事件发生时间
槽 2：事件类型
槽 3：事件种类
槽 4：犯罪者
侧面 4-1：个人
侧面 4-2：组织
侧面 4-3：置信度
槽 5：实物目标
侧面 5-1：名称列表
侧面 5-2：数量
侧面 5-3：类型
槽 6：人目标
侧面 5-1：姓名列表
侧面 5-2：数量
侧面 5-3：类型
槽 7：事件发生地点
槽 8：对实物目标的影响
槽 9：对人目标的影响

利用“恐怖袭击事件”框架，句子“在位于巴黎 11 区的巴塔克兰剧院，多名武装分子在巴黎当地时间 13 日晚劫持了正在剧院观看演出的大约 1500 名观众并与警方展开对峙。”的语义可以表示为：

事件发生时间：巴黎当地时间 13 日晚
事件类型：劫持
事件种类：恐怖袭击
人目标：
类型：观众
数量：约 1500

事件发生地点：巴黎 11 区的巴塔克兰剧院

框架表示方法可以有效的表达结构性语义，并能够讲语义的内部结构关系及语义间的联系表示出来。此外，框架表示法可以将槽位设置为另一个框架，从而实现框架间的联系，构建更加复杂的框架网络，还可以实现框架之间的继承关系。但是，精确表达语义需要非常复杂详细的框架以及很多嵌套层级的框架结构。

4.2.3 语义网表示法

语义网络（Semantic Network）是一种用实体及其语义关系来表达知识和语义的网络图。语义网络由节点和弧线组成：节点表示各种事件、事物、概念、属性、动作等，也可以是一个语义子网络；弧线表示节点之间的语义关系，并且是有方向和标注的，方向表示节点间的主次关系且方向不能随意调换。图4.1给出了“大学”的语义网表示样例。

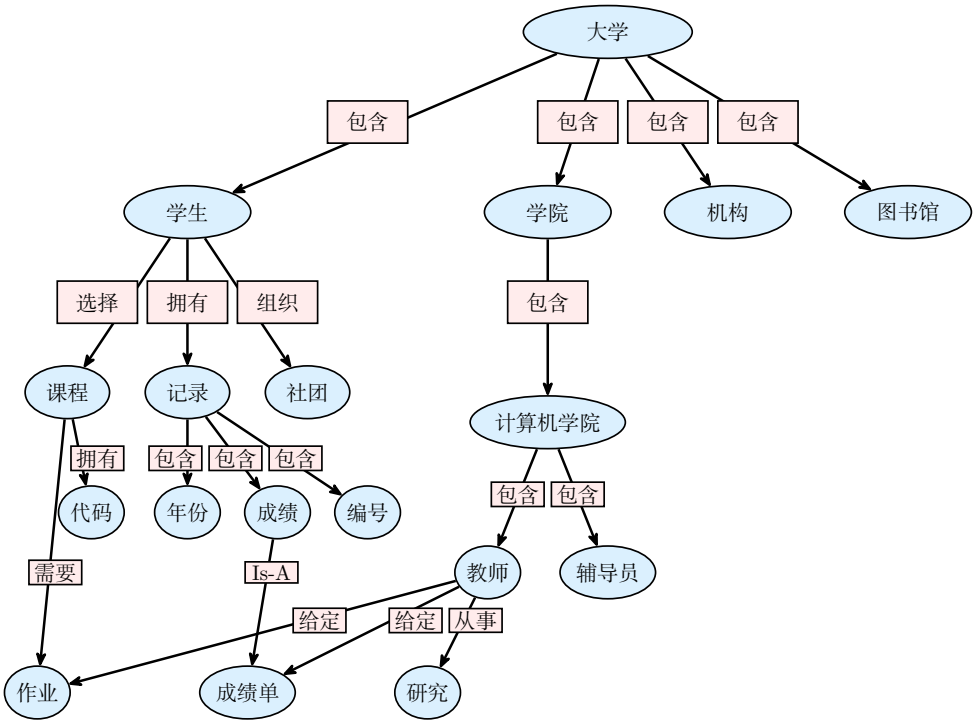

图 4.1 “大学”的语言网表示样例

语义网除了可以描述事物间包括类属关系、聚集关系、时间关系、位置关系、推论关系等多种复杂语义关系外，还可以通过增加节点的方法表示合取、析取、蕴含等语义表示中常用的连接

词。例如，句子“如果明天下雨，就去看电影或者唱歌”的语义网表示如图4.2所示。

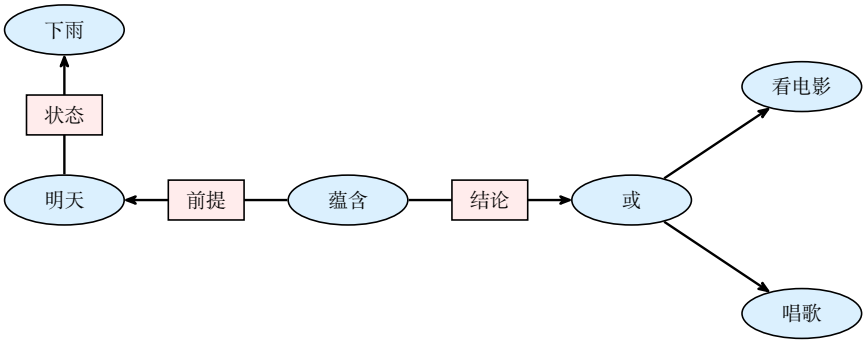

图 4.2 “如果明天下雨，就去看电影或者唱歌”的语言网表示样例

对于比较复杂的语义还能涉及“每一个”、“有一个”等量词时，使用语义网进行表示时可以通过引入分区技术进行实现[10]。其基本思想是将复杂的语义划分为若干个子语义，每个子语义采用语义网进行表示。若干个子网络合并构成更大的网络。语义网可以逐层嵌套，子网络之间也可以采用弧线进行连接。例如：“所有的学生都完成了一个课程设计”的语义网表示如图4.3所示。

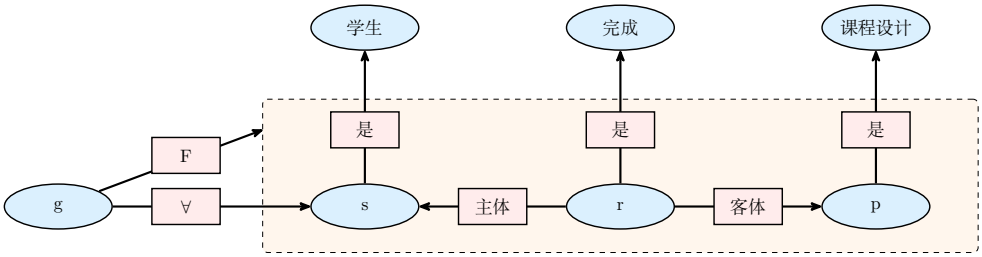

图 4.3 “所有的学生都完成了课程设计”的语言网表示样例

图4.3中节点“s”、“r”、“p”构成了子语义网，其中“s”是全称变量、“r”和“p”是存在变量。节点“g”是表示这个子网络，由弧线“F”指向其所代表的子网络结构。

语义网可以较好的把事务的属性以及事务之间的各种语义联系显式的进行表示，也可以较容易的实现语义检索。但是，由于语义网没有公认的形式表示体系，所表达的语义需要依赖分析算法如何对其进行解释，表示形式的不唯一又进一步增加了其处理的复杂性。

4.3 分布式表示

分布式表示（Distributed Representation）旨在将文本表示为低维空间下稠密的向量，并在低维表示空间中利用表示向量之间的计算关系，体现文本间的语义关联。现代的文本表示学习，很大程度上受到 Salmon 等人在 1975 年提出的向量空间模型（Vector Space Model）[11] 的影响。这项工作结合信息检索领域的具体应用，阐述了将单词和篇章表示为向量的思想。一方面，对文本的处理可以直观地映射到向量空间，体现为对文本向量的加法、减法、距离度量等操作；另一方面，可以将向量化的文本作为输入，从而直接将统计学习与机器学习算法应用在自然语言处理应用上。结合语言模型的相关理论，向量化文本的思想在统计学习时期就已在自然语言处理系统中广泛应用。

文本分布式表示进一步提升了向量化文本的实用性，使文本表示模块成为自然语言处理系统必不可少的一部分。在分布式表示提出之前，许多自然语言处理算法采用的独热表示（One-hot Representation）作为机器学习模型的输入。独热表示的维度和词表的大小一致，存在表示稀疏性的问题，而且无法表示单词之间的语义相似度。分布式表示通过将文本表示为低维空间下稠密的向量，有效地解决了这一问题。当应用在下游任务时，文本分布式表示也体现出良好的泛化能力，而且能有效地编码任务所需要的语法和语义信息[12, 13]。因此，文本分布式表示作为模型的基本输入，已经被广泛应用于自然语言处理领域的各种任务。

早期的分布式表示方法聚焦在词汇的表示向量构建上。随着自然语言处理技术应用领域的拓展，为了适配多样化的任务需求，句子、篇章级别的分布式表示方法也逐渐广泛应用。本节针对不同级别的语言粒度，介绍单词、句子和篇章级别分布式表示方法。

4.3.1 单词分布式表示

单词分布式表示（Word Distributed Representation）通过将单词表示为定长低维稠密向量，在向量空间建构单词之间的语义关系。形式上，单词分布式表示的目标是建立单词嵌入矩阵 $\boldsymbol{W} \in \mathbb{R}^{|\mathbb{V}|*d}$，其中矩阵的每一行对应一个单词，为单词的向量表示，即词向量。

例如：

“计算机”表示为 [0.16, 0.19, -0.28, ..., 0.87]
“电脑”　表示为 [0.20, 0.17, -0.21, ..., 0.97]
“冰激凌”表示为 [-0.90, 0.72, 0.65, ..., 0.06]

相比于独热表示，分布式表示可以编码不同单词之间的语义关联。如上例中，如果采用独热表示，“计算机”与“电脑”以及“计算机”与“冰激凌”之间的相似度都相同。但是采用分布式表示可以使得“计算机”和“电脑”在大多数维度上相近，这样“计算机”和“电脑”的向量之间的距离可以远小于“计算机”和“冰激凌”之间的距离。

根据单词分布式表示的目标，即在向量空间建构单词之间的语义关联，使含义相近的单词具有相似的向量表示。这自然地引出了两个问题：（1）如何衡量单词含义的相近；（2）如何衡量表示的相似。针对第一个问题，大部分单词分布式表示方法遵从分布式假设，即出现在相同上下文

中的单词往往具有相似的含义[14]。在分布式假设的基础上，这些方法侧重于还原单词之间的共现关系，即为频繁出现在相同上下文中的词语之间赋予较高的表示相似度。针对第二个问题，根据下游应用场景的不同，可以根据表示向量的余弦相似度、L2 范数距离等方式衡量表示向量的相似性。为了实现上述的目标，早期的词向量模型通过统计方法，根据文本中词级别信息的统计数据构建低维稠密词向量。目前更多的方式是利用机器学习方法，通过对大量无标注文本进行自监督学习，使用机器学习模型来学习词向量。

本节将分别介绍基于统计和机器学习的词向量模型。

1. 基于共现矩阵奇异值分解的词向量模型

在分布式假设下，希望单词之间的相似度体现为两词出现在相同上下文的频率。因此可以采用针对共现矩阵（Co-occurrence Matrix）的矩阵分解方法。隐式语义分析（Latent Semantic Analysis，LSA）模型[15] 采用奇异值分解方法（Singular Value Decomposition，SVD），将单词文档共现矩阵（Term-Document Co-occurrence Matrix）或单词上下文共现矩阵（Window based Co-Occurrence Matrix）转换为单词向量表示。本节以单词上下文共现矩阵为例，介绍基于奇异值分解的词向量模型。

共现矩阵 $\boldsymbol{A} \in R^{|\mathbb{V}|\times|\mathbb{V}|}$，$\boldsymbol{A}_{ij}$ 表示词表 $\mathbb{V}$ 中下标为 i 和 j 的单词出现在相同上下文中的次数。根据语料库，可以根据算法4.1构建共现矩阵，用于后续词表示的构建。对于句子中的单词，将其向前、向后各 n 个单词的范围作为该单词的上下文范围，称为该单词大小为 n 的上下文窗口。对于每个句子，取其中每个单词大小为 n 的上下文窗口，对窗口范围内的每个单词与该单词进行共现计数。对于语料库整体进行共现次数统计，获得共现矩阵。

共现矩阵的每一行可以自然地当做对应词的向量表示，因为它的不同维度表示和各个单词的共现次数，共现矩阵提供的向量表示可以体现词语之间的相似度。然而，共现矩阵提供的词向量表示维度和词表大小相同，依然面临表示稀疏性的问题。奇异值分解方法将共现矩阵分解为低阶近似矩阵，从而为每个单词提供低维表示。对于矩阵 $\boldsymbol{A} \in \mathbb{R}^{m\times n}$，通过奇异值分解可将其分解为 $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$，其中 $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ 和 $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ 是酉矩阵，$\boldsymbol{\Sigma} \in \mathbb{R}^{m\times n}$ 在主对角线之外的元素为 0。特征向量的顺序均按特征值由大及小顺序排列。矩阵 $\boldsymbol{\Sigma}$ 对角线上的奇异值元素均为非负实数，且按由大及小顺序排列。

一般情况下，矩阵奇异值的大小分布极不均匀，前 10% 奇异值之和可以占到全部奇异值之和的 90% 以上。因此，可以在奇异值矩阵中只保留最大的 d 个奇异值，同时在两侧矩阵仅保留对应的分量，对矩阵 $\boldsymbol{A}$ 进行低秩近似。应用上述方法，对于对共现矩阵进行数据压缩，就获得单词的低维稠密表示 $\boldsymbol{W} \in \mathbb{R}^{|V|\times d}$，目的是使 $\boldsymbol{W}_i\boldsymbol{W}_j^T$ 能够近似地还原词表中下标为 i, j 的单词在相同上下文中出现的频率。词表示矩阵 $\boldsymbol{W}$ 的计算如算法4.2所示。

相对于词语的独热表示，基于共现矩阵奇异值分解的词向量模型初步解决了表示稀疏性的问题，且可以在一定程度上体现词语之间的相似度。

代码 4.1: 共现矩阵统计算法

输入: 训练语料库 D，上下文窗口大小 n
输出: 词表 $\mathbb{V}$，共现矩阵 A

```
𝕍 = unique(D)                                   // 由语料库统计出现的词表
A_ij = 0, ∀i, j ∈ [1, ..., |𝕍|]                 // 初始化共现矩阵
foreach s ∈ D do
    w_1, ..., w_N = s                           // 取训练语料库中的句子，句子由 N 个词构成
    id_i = 𝕍.index(w_i), ∀i ∈ [1, ..., N]       // 将句子中的单词转化为词表下标
    for i = 1 to N do
        for j = max(1, i − n) to min(N, i + n), j ≠ i do
            A_{id_i, id_j} = A_{id_i, id_j} + 1  // 共现计数
        end
    end
end
return 𝕍, A
```

代码 4.2: 基于共现矩阵奇异值分解的词向量模型[16]

输入: 共现矩阵 $\boldsymbol{A} \in \mathbb{R}^{|\mathbb{V}| \times |\mathbb{V}|}$，嵌入维度 d
输出: 词表示矩阵 $\boldsymbol{W} \in \mathbb{R}^{|\mathbb{V}| \times d}$

```
U, Σ, V = SVD(A)                    // 由 A 是对称方阵，可知 U = V，Σ 是对角方阵
U, Σ = U_{:,:d}, Σ_{:d,:d}          // 特征降维，保留最重要的 d 维，A ≈ UΣU^T
diag(σ_1, ..., σ_d) = Σ             // Σ 包含最重要的 d 个奇异值 {σ_i}_{i=1}^d
√Σ = diag(√σ_1, ..., √σ_d)          // √Σ √Σ^T = Σ
W = U√Σ                             // WW^T = UΣU^T ≈ A return W
```

2. 基于上下文单词预测词向量模型

Mikolov 等人在文献 [17] 中提出了大幅度简化以往的神经网络语言模型（Neural Probabilistic Language Model, NPLM）的方法，去除了非线性隐藏层，使用自监督的方式从大量无监督文本训练词表示模型。构建了两个非常简单的神经网络模型结构：连续词袋模型（Continuous Bag Of Words, CBOW）和跳字模型（Skip-Gram，SG），用于学习单词分布式表示。其框架如图4.4和图4.5和所示。它们分别基于不同的假设，以基于条件概率的方式训练词表示模型。另外，针对梯度反向传播计算量过大的问题，采用负采样（Negative Sampling）和层次 Softmax（Hierarchical Softmax）两种近似训练方法。

Sikp-Gram 模型的基本假设是文本中的词可以预测其上下文窗口内的其他词。如图4.4所示，在句子“我稍后回答这个问题”中，对于中心词“回答”，当以大小为 2 的上下文窗口预测时，模型考虑

上下文窗口内词语“我”，“稍后”。“这个”，“问题”在中心词为“回答”条件下的出现概率。Skip-Gram 模型通过最大化上下文词在给定中心词时的条件概率，来进行模型的训练。

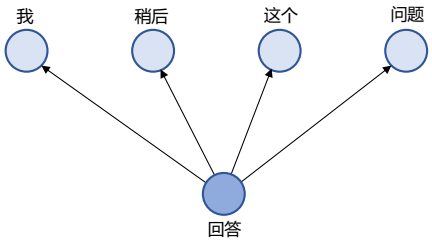

图 4.4　Skip-Gram 模型对上下文词的示例

Skip-Gram 模型以负对数概率形式的损失函数作为优化目标，形式化表示为：

$$\mathcal{L}(w_1 \dots w_T) = -\sum_{t=1}^{T} \sum_{-n \leqslant i \leqslant n, i \neq 0} \log P(w_{t+i}|w_t) \tag{4.1}$$

其中，w_t 是位置 t 的中心词，w_{t+i} 是上下文窗口内每个位置的上下文词，T 是文本序列的长度。$P(w_{t+i}|w_t)$ 条件概率通过通过训练上下文词和中心词的词嵌入参数矩阵，来近似估计上述条件概率。具体来说，Skip-Gram 包括 $\boldsymbol{U} \in \mathbb{R}^{|\mathbb{V}| \times d}$ 和 $\boldsymbol{V} \in \mathbb{R}^{|\mathbb{V}| \times d}$ 两个词嵌入矩阵，分别表示词表中每个单词作为上下文词和中心词时的词向量。Skip-Gram 模型通过上下文词和中心词向量的相似度估计上下文词的出现概率，具体公式如下所示：

$$P(w_o|w_c) = \frac{\exp(\boldsymbol{u}_o^T \boldsymbol{v}_c)}{\sum_{i \in \mathbb{V}} \exp(\boldsymbol{u}_i^T \boldsymbol{v}_c)} \tag{4.2}$$

其中，w_o 和 w_c 分别表示特定的上下文词和中心词，$\boldsymbol{u}_o$ 是 w_o 用作上下文词的表示，$\boldsymbol{v}_c$ 是 w_c 用作中心词的表示，$\boldsymbol{u}_i$ 是词表中每个词用作上下文词的表示。在优化上述目标函数后，Skip-Gram 模型通常采用训练好的中心词表示作为最终的词表示。

CBoW 模型则假设文本中的词可以通过其在文本中的上下文词推导出来。如图4.5所示，在句子“我稍后回答这个问题”中，当以大小为 2 的上下文窗口预测中心词“回答”时，模型仅考虑上下文词“我”，“稍后”,“这个”，“问题”，以此推断中心词的出现概率。CBoW 模型通过最大化中心词在上下文中出现的条件概率，来进行模型的训练。

CBoW 模型也是以负对数概率形式的损失函数作为优化目标：

$$\mathcal{L}(w_1 \dots w_T) = -\sum_{t=1}^{T} \log P(w_t|w_{t-n}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+n}) \tag{4.3}$$

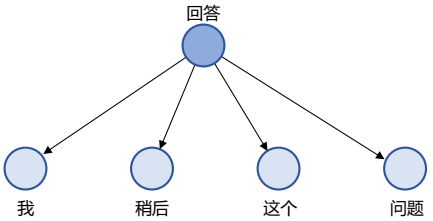

图 4.5 CBoW 模型对中心词的预测示例

CBoW 模型也通过学习上下文词向量矩阵和中心词词向量矩阵来优化目标函数。通常使用与 Skip-Gram 相反的记号，即用 $\boldsymbol{U} \in \mathbb{R}^{|\mathbb{V}| \times d}$ 表示中心词词向量矩阵，$\boldsymbol{V} \in \mathbb{R}^{|\mathbb{V}| \times d}$ 表示上下文词向量矩阵。根据上下文词生成中心词的条件概率具体计算公式如下所示：

$$\boldsymbol{v}_o = \frac{1}{2m} \sum_{-n \leqslant i \leqslant n, i \neq 0} \boldsymbol{v}_{c+i} \tag{4.4}$$

$$P(w_c | w_{c-n}, \ldots, w_{c-1}, w_{c+1}, \ldots, w_{c+n}) = \frac{exp(\boldsymbol{u}_c^T \boldsymbol{v}_o)}{\sum_{i \in \mathbb{V}} exp(\boldsymbol{u}_i^T \boldsymbol{v}_o)} \tag{4.5}$$

其中，$\boldsymbol{v}_o$ 是平均的上下文词向量，用于计算和中心词的相似度；$\boldsymbol{u}_c$ 是 w_c 用作中心词的表示，$\boldsymbol{u}_i$ 是词表中每个词用作中心词的表示。

在实际应用中，由于词表内通常包含数万甚至数十万单词，Skip-Gram 和 CBoW 模型在基于 Softmax 计算上下文词和中心词的出现概率进行梯度更新时，会产生非常大规模的计算开销。因此，通常使用负采样或者层次 Softmax 的方法降低计算开销。下面我们结合 Skip-Gram 模型介绍这些上述两种方法。

在 Skip-Gram 的目标函数中，如公式4.1所示，优化的最终目标是最大化上下文词的条件概率 $P(w_{t+i}|w_t)$，即 $P(w_o|w_c)$。结合公式4.2来看，这个损失函数使正确的上下文词 w_o 和中心词 w_c 之间的相似度 $\boldsymbol{u}_o^T \boldsymbol{v}_c$ 尽量大，同时对于词表中每个不在上下文窗口内的词，使其和中心词的相似度 $\boldsymbol{u}_i^T \boldsymbol{v}_c$ 尽量小。后者针对词表中每个词进行计算，因此引入大量的计算开销。针对上述问题，**负采样**（Negative Sampling）将目标函数中全体词表范围的相似度计算修正为目标词和 K 个负例的相似度计算，其中 K 是远小于词表大小的超参数。通过这种方式，使得训练的计算开销与词表大小无关，而只与超参数 K 相关。

具体地，首先将词向量的相似度和词出现在上下文窗口的概率相关联：

$$P(D = 1 | w_o, w_c) = \sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c) \tag{4.6}$$

$$P(D = 0 | w_o, w_c) = 1 - P(D = 1 | w_o, w_c) \tag{4.7}$$

其中，$D=1$ 或 $D=0$ 表示这组上下文词-中心词是否来自训练语料，即训练语料中单词 w_o 是否出现在 w_c 的上下文窗口中。σ 表示 Sigmoid 函数，$\boldsymbol{u}_o$ 是 w_o 用作上下文词的表示，$\boldsymbol{v}_c$ 是 w_c 用作中心词的表示。

在优化目标方面，对于每组上下文词-中心词，原始的 Skip-Gram 模型只优化 $P(D=1|w_o,w_c)$ 的正样本,而负采样下的模型从原始词分布 $P(w)$ 中采样 K 个未出现在上下文窗口中的词 $w_1,\ldots,w_K$ 作为负样本，并额外将 $P(D=0|w_i,w_c)$ 作为优化目标，形式化表示为：

$$P(w_o|w_c)=P(D=1|w_o,w_c)\prod_{i=1}^{K}P(D=0|w_i,w_c) \tag{4.8}$$

将上述公式代替公式4.2中的概率项估计，并代入公式4.1，可以得到负采样下的目标函数。

相比于负采样，层次 Softmax 则以另外一种角度为上下文词的出现概率 $P(w_o|w_c)$ 建立近似训练方法。如图4.6所示，树的每个叶子结点表示词表中的一个词语。用每个分叉结点 n 的上下文表示向量 $\boldsymbol{u}_n$ 建模其左、右子树上每个词语出现在上下文窗口中的概率之和。

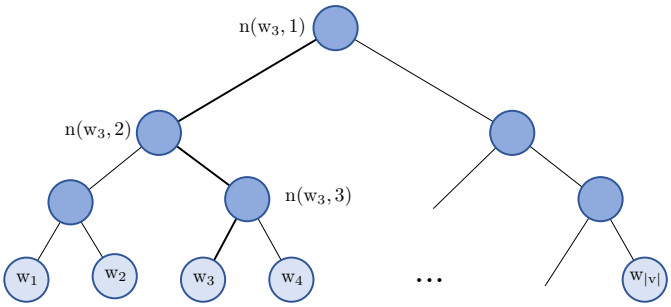

图 4.6　层次 softmax 将词表组织成二叉树结构，树的每个叶子结点代表词表中的一个词语

具体地，可以使用下述公式对此概率进行估计：

$$\sum_{w\in \mathrm{lSub}(n)}\frac{P(w|w_c)}{\sum_{w\in n}P(w|w_c)}=\sigma(\boldsymbol{u}_n^T\boldsymbol{v}_c) \tag{4.9}$$

其中，$\mathrm{lSub}(n)$ 表示对应的叶子结点位于 n 的左子树下的全体单词集合。

由于 $\sigma(x)+\sigma(-x)=1$，可以得到：

$$\sum_{w\in \mathrm{rSub}(n)}\frac{P(w|w_c)}{\sum_{w\in n}P(w|w_c)}=1-\sigma(\boldsymbol{u}_n^T\boldsymbol{v}_c)=\sigma(-\boldsymbol{u}_n^T\boldsymbol{v}_c) \tag{4.10}$$

类似地，rSub(n) 表示对应的叶子结点位于 n 的右子树下的全体单词集合。

在公式4.9和4.10的基础上，对于词表中的词 w，用 $L(w)$ 表示从树的根结点到词 w 对应的叶子结点的路径，其中包括从根结点到叶子结点的父结点的全部非终结点，但不包括叶子结点本身。记 $n(w,i)$ 是路径 $L(w)$ 上的第 i 个结点，其上下文表示向量记为 $\boldsymbol{u}_i$，则词 w 出现在 w_c 的上下文窗口中的概率估计为：

$$P(w|w_c) = \prod_{i=1}^{|L(w)|} \sigma(\boldsymbol{u}_i^T \boldsymbol{v}_c) \tag{4.11}$$

可以发现，对于任意的中心词 w_c，层次 Softmax 保证词表中所有词出现在中心词上下文窗口特定位置的条件概率之和为 1，即 $\sum_{w\in\mathbb{V}} P(w|w_c) = 1$。基于层次 softmax 计算词表示模型，计算开销和 $|L(w)|$ 的平均值呈线性关系。当使用满二叉树结构容纳词表时，$|L(w)|$ 具有 $\mathcal{O}(\log_2 |\mathbb{V}|)$ 的上界，所以层次 Softmax 也可以显著降低 Skip-Gram 模型的计算复杂度。

3. 全局向量（GloVe）模型

上一节中介绍的 Word2Vec 词表示模型利用每个单词的上下文窗口信息作为监督信号，自监督地对语料库进行学习。除此之外，使用的词共现矩阵训练的隐式语义分析模型指出，语料库的全局统计信息对词表示学习同样起到关键的作用。全局向量（Global Vectors for Word Representation，GloVe）模型[18] 结合了上述模型的思想，从共现概率的角度分析并改进了 Skip-Gram 模型，既使用文本中局部的上下文信息，又对语料库的全局共现统计数据加以利用。

GloVe 模型基于上下文窗口共现矩阵的统计，即对语料库中特定中心词-上下文词对的出现次数的统计。在算法4.1所述的共现计数方法基础上，GloVe 模型中的共现矩阵进一步地考虑了中心词和上下文词之间的距离，使相距更近的中心词-上下文词对为共现次数起到更大的贡献。记 w_i, w_j 为词表中下标为 i, j 的单词，在它们的每次共现中，记 $d(w_i, w_j)$ 为单词 w_i, w_j 之间的距离。GloVe 模型中的共现矩阵将词与词之间的共现次数按共现距离的倒数进行加权，即 $C_{ij} = \sum d^{-1}(w_i, w_j)$。由共现矩阵可以得到单词 w_j 出现在单词 w_i 上下文的共现概率为 $p_{ij} = P(w_j|w_i) = \frac{C_{ij}}{\sum_{j=1}^{|\mathbb{V}|} C_{ij}}$.

GloVe 模型的损失函数形式和先前介绍的 Word2Vec 模型相似，同样以还原共现频率 p_{ij} 为目标，并在其基础上进行改进。记 $\hat{p}_{ij} = P(w_j|w_i)$ 为 w_j 出现在 w_i 上下文范围内的预测概率，即 p_{ij} 的预测值。在共现矩阵统计的视角下，Skip-Gram 模型使用统计的共现频率作为监督信号，通过估计共现概率优化词向量。此时，Skip-Gram 模型的损失函数可以重新表示如下：

$$\mathcal{L} = -\sum_{i=1}^{|\mathbb{V}|}\sum_{j=1}^{|\mathbb{V}|} C_{ij} \log \hat{p}_{ij} \propto -\sum_{i=1}^{|\mathbb{V}|} x_i \sum_{j=1}^{|\mathbb{V}|} p_{ij} \log \hat{p}_{ij} \tag{4.12}$$

其中

$$\hat{p}_{ij} = \frac{\exp(\boldsymbol{u}_j^T \boldsymbol{v}_i)}{\sum_{\mathrm{j}=1}^{|\mathbb{V}|} \exp(\boldsymbol{u}_\mathrm{j}^T \boldsymbol{v}_i)} \tag{4.13}$$

x_i 是训练语料中 w_i 作为中心词出现的次数，正比于训练语料中 w_i 的出现频率。

通过公式4.12，可以将 Skip-Gram 模型理解为使用交叉熵损失，优化词共现条件概率分布的过程。GloVe 模型对 Skip-Gram 模型进行了如下改进。首先，GloVe 模型用平方损失代替 Skip-Gram 模型中的交叉熵损失，并使用变量 $p'_{ij} = C_{ij}$ 和 $\hat{p}'_{ij} = \exp(\boldsymbol{u}_j^T \boldsymbol{v}_i)$ 代替原来的概率分布。另外，为每个单词 w_i 引入可训练的中心词偏置项 b_i 以及上下文词偏置项 c_i 对训练目标进行校正。在对数形式下，平方损失项如下所示：

$$(\log p'_{ij} - \log \hat{p}'_{ij})^2 = (\boldsymbol{u}_j^T \boldsymbol{v}_i - \log C_{ij} + b_i + c_j)^2 \tag{4.14}$$

此外，GloVe 模型使用 $h_{ij} = h(C_{ij})$ 作为每个损失项的权重，建模单词 w_i 与 w_j 的相关度。对于 h_{ij}，目标是为共现频率较高的词对赋予较高的权重，所以 h 是 C_{ij} 的非递减函数。此外，不希望这个权重随共现频率无限地增大，故当函数值到达界限之后不应继续增加。另外，对于从未共现的单词 w_i 与 w_j，应有 $h_{ij} = 0$，表示两者之间不存在关联。综上所述，使用如下分段函数来建模每个损失项的权重：

$$h(c) = \begin{cases} (c/c_{max})^\alpha, & 0 \leqslant c < c_{max} \\ 1, & c \geqslant c_{max} \end{cases} \tag{4.15}$$

其中，c_{max} 和 α 是预设的超参数。

GloVe 模型最终的损失函数如公式4.16所示。在训练中，为了提高训练效率，可以省略任意 $h_{ij} = 0$ 的损失项，例如在每步优化时随机抽取一批次 $C_{ij} > 0$ 的词对进行梯度更新。

$$\mathcal{L} = -\sum_{i=1}^{|\mathbb{V}|} \sum_{j=1}^{|\mathbb{V}|} h_{ij} (\boldsymbol{u}_j^T \boldsymbol{v}_i - \log C_{ij} + b_i + c_j)^2 \tag{4.16}$$

考虑到共现矩阵为对称矩阵，即 $C_{ij} = C_{ji}$，在 GloVe 模型中优化得到的中心词向量和上下文词向量理论上是相等的。在实际应用中，由于权重的随机初始化不同，同一个词最终得到的中心词向量和上下文词向量可能不相等。GloVe 通常使用两个向量的和作为输出的词向量。

4. 基于字节对编码的子词表示模型

本章前几节所介绍的词表示模型都依赖预先确定的词表 $\mathbb{V}$，在编码输入词序列时，这些词表示模型只能处理词表中存在的词。因此，在使用中，如果遇到不在词表中的未登录词，模型无法为其生成对应的表示，只能给予这些未登录词一个默认的通用表示。通常的处理方式是，词表示模型会预先在词表中加入一个默认的“[UNK]”（unknown）标识，表示未知词，并在训练的过程中将

[UNK] 的向量作为词表示矩阵的一部分一起训练，通过引入某些相应机制来更新 [UNK] 向量的参数。在使用时，对于全部的未登录词，使用 [UNK] 的向量作为这些词的表示向量。基于固定词表的词表示模型对词表大小的选择比较敏感。当词表大小过小时，未登录词的比例较高，影响模型性能。而当词表大小过大时，大量低频词出现在词表中，而这些词的词向量很难得到充分学习。理想模式下，词表示模型应能覆盖绝大部分的输入词，并避免词表过大所造成的数据稀疏问题。

为了缓解未登录词问题，一些工作通过利用亚词级别的信息构造词表示向量。一种直接的解决思路是为输入建立字符级别表示，并通过字符向量的组合来获得每个单词的表示，以解决数据稀疏问题。然而，单词中的词根、词缀等构词模式往往跨越多个字符，基于字符表示的方法很难学习跨度较大的模式。为了充分学习这些构词模式，子词表示模型提出了子词（Subword）的概念，试图缓解上文介绍的未登录词问题。子词表示模型会维护一个子词词表，其中既存在完整的单词，也存在形如“c”, “re”, “ing”等单词部分信息，称为**子词**。子词表示模型对词表中的每个子词计算一个定长向量表示，供下游模型使用。对于输入的词序列，子词表示模型将每个词拆分为词表内的子词，例如将单词“reborn”拆分为“re”和“born”. 模型随后查询每个子词的表示，将输入重新组成为子词表示序列。当下游模型需要计算一个单词或词组的表示时，可以将对应范围内的子词表示合成为需要的表示。因此，子词表示模型能够较好地解决自然语言处理系统中未登录词的问题。

本节以字节对编码模型（Byte Pair Encoding，BPE）[19] 为例，介绍子词表示模型。BPE 该模型所采用的词表包含最常见的单词以及高频出现的子词。在使用中，常见词通常本身位于 BPE 的词表中，而罕见词通常能被分解为若干个包含在 BPE 词表中的子词，从而大幅度降低未登录词的比例。BPE 算法包括两个部分：（1）子词词表的确定；（2）全词切分为子词以及子词合并为全词的方法。下面分别介绍上述的两个部分。

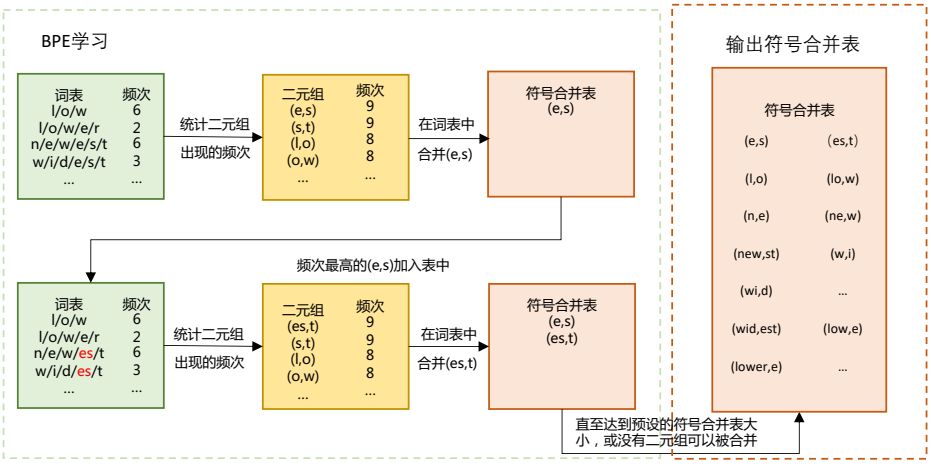

图 4.7 BPE 模型中子词词表的计算过程

BPE 词表的计算过程如图4.7所示。首先确定语料库中全词的词表和词频，然后将每个单词切分为单个字符的序列，并在序列最后添加符号“</w>”作为单词结尾的标识。比如单词“low”被切分为序列 “l␣o␣w␣</w>”。所切分出的序列元素称为字节，即每个单词都切分为字节的序列。之后，按照每个字节序列的相邻字节对和单词的词频，统计每个相邻字节对的出现频率，合并出现频率最高的字节对，将其作为新的子词加入词表，并将全部单词中的该字节对合并为新的单一字节。如图4.7所示，在第一次迭代时，出现频率最高的字节对是 (e,s)，故将 “es” 作为子词加入词表，并将全部序列中相邻的 (e,s) 字节对合并为 es 字节。重复这一步骤，直至 BPE 子词词表的大小达到指定的预设值，或没有可合并的字节对为止。

在子词词表确定之后，对于输入词序列中未在词表中的全词进行切分，BPE 算法对词表中的子词按从长到短的顺序进行遍历，用每一个子词和当前序列中的全词或未完全切分为子词的部分进行匹配，将其切分为该子词和剩余部分的序列。如图4.8所示，对于全词”lowest</w>”，首先通过匹配子词”est</w>” 将其切分为”low”, ”est</w>” 的序列，再通过匹配子词”low”，确定其最终切分结果为”low”, ”est</w>” 的序列。通过这样的过程，BPE 尽量将词序列中的词切分成已知的子词。

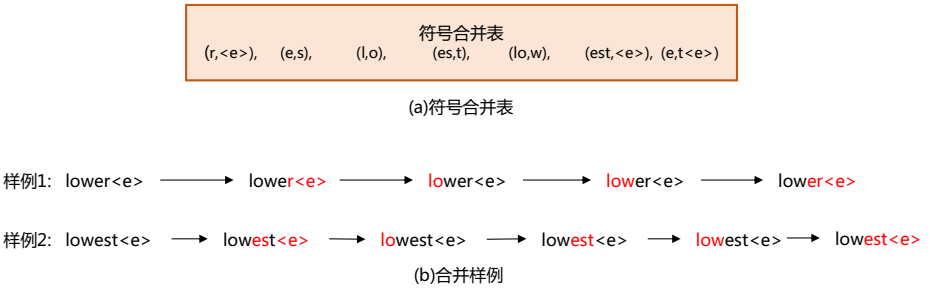

图 4.8　BPE 模型子词切分和合并示例

在遍历子词词表后，对于切分得到的子词序列，为每个子词查询子词表示，构成子词表示序列。若出现未登录子词，即未出现在 BPE 词表中的子词，则采取和未登录词类似的方式，为其赋予相同的表示，最终获得输入的子词表示序列。

对于使用了子词表示模型的自然语言处理系统，比如机器翻译系统，其输出序列也是子词序列。如图4.8所示，对于原始输出，根据终结符 </w> 的位置确定每个单词的范围，合并范围内的子词，将输出重新组合为词序列，作为最终的结果。

5. 单词分布式表示的评价与应用

单词分布式表示模型的定量评估方法主要分为内部（Intrinsic Evaluation）和外部（Extrinsic Evaluation）方法。内部评估方法通常基于一个特殊设计的辅助任务，这个辅助任务探测词向量应该具有的某种性质，如词义相关性、类比性等，并最终返回一个分数，来表示词向量的好坏，从而

帮助我们理解词向量模型的特点。外部评估方法通常基于一个实际应用任务，通过将词向量作为该任务的输入表示，比较不同词向量模型在该任务上的性能，来选择适合于该任务的词向量模型。在评价词向量模型的综合性能时，通常会使用内部评估方法。一方面，内部评估方法所使用的辅助任务也比一般的应用任务简单，且计算速度快。另一方面，外部评估方法除了词向量模型外所，还会涉及下游任务的模型。当系统整体表现达不到预期时，问题可能来自于词向量模型，也可能来自任务模型或两者之间的交互，因此不能很好地指导词向量模型的选择与改进。本节介绍两种常用的内部评估方法，即词义相关性和词语类比性。

词义相关性任务通过探索词向量对词义相关性的表达能力，来评价词向量的质量。理想状态下，词向量应该稠密、连续地分布在低维语义空间上，所以应该存在一种形式简洁、易于计算的相似度度量，使得词向量之间的相似度可以反映词语之间的词义相关性。一般地，对于单词 w_i, w_j 及其词向量 $\boldsymbol{v}_i, \boldsymbol{v}_j$，简单地使用余弦相似度作为词义相似性的度量：

$$\text{sim}(w_i, w_j) = \cos(\boldsymbol{v}_i, \boldsymbol{v}_j) = \frac{\boldsymbol{v}_i \boldsymbol{v}_j}{\|\boldsymbol{v}_i\| \|\boldsymbol{v}_j\|} \tag{4.17}$$

通过直接将词义相似度作为目标，可以定量衡量词向量模型的性能。常用的英文词义相似度评测基准包括 WordSim-353 [20] 和 SimLex-999 [21]，他们分别包含 353 和 999 个英文词对，每个词对包括一个位于 [0, 10] 区间内的相似度分数，如表4.5所示。针对词向量模型的评测，由词向量计算的语义相似度与标注值之间的 Spearman 或 Pearson 相关系数可以作为词向量模型的评价。

表 4.5　语义相似度评测基准样本示例

单词 1	单词 2	相似度
dirty	narrow	0.30
student	pupil	9.40
win	dominate	5.68
smart	dumb	0.60
attention	awareness	8.73
leave	enter	1.38

在应用中，对于给定的单词 w，可以通过词向量的余弦相似度，在词表中检索意义最为接近的词语，即 $w^* = \arg\max_{w' \in \mathbb{V}} \text{sim}(w, w')$。另外，词向量还可以应用于类比性任务。例如，在由 (”man”, ”woman”) 词对确定的类比关系下，可以为单词”son” 检索类比词”daughter”，它们满足“man 之于 woman，相当于 son 之于 daughter”的类比关系。一般地，对于形如“w_a 之于 w_b，相当于 w_c 之于 ?”的问题，我们可以通过以下公式检索最恰当的类比词：

$$w^* = \arg\min_{w' \in \mathbb{V}} \cos(\boldsymbol{v}^*, \boldsymbol{v}_c + \boldsymbol{v}_b - \boldsymbol{v}_a) \tag{4.18}$$

其中 $\boldsymbol{v}^*, \boldsymbol{v}_a, \boldsymbol{v}_b, \boldsymbol{v}_c$ 分别是 w^*, w_a, w_b, w_c 对应的词向量。

4.3.2 句子分布式表示

句子分布式表示主要用于句子级别的任务，如情感分析、文本推理、语义匹配等。对于句子级别表示的构建，不但要考虑句子中所包含单词的语义，也要考虑句子内部词之间的关系，即词的共现信息和句子语义之间的联系；以及句子和句子之间隐含的语义相似性，以及其他的语义关系。这些性质对于句子级别的下游应用任务都很重要。本节将介绍两种句子级分布式表示算法 Skip-Thought 和 Sent2Vec 模型。

1. Skip-Thought 句子表示模型

Skip-Thought 模型[22] 的目的主要是建模句子与句子之间的上下文语义关系。Skip-Thought 模型借鉴了 Skip-Gram 模型的思想，认为可以基于一个句子预测出其上下文的句子，并以此作为监督信号，学习句子之间的语义关系，得到句子表示模型。Skip-Thought 模型结构如图4.9所示，包括一个编码器和两个结构相同的解码器，输入当前位置的句子 $s^i = w_1^i, \ldots, w_N^i$，编码器将输入句转化为表示向量，而两个解码器分别预测该句在上下文中的前一个句子 s^{i-1} 和后一个句子 s^{i+1}。

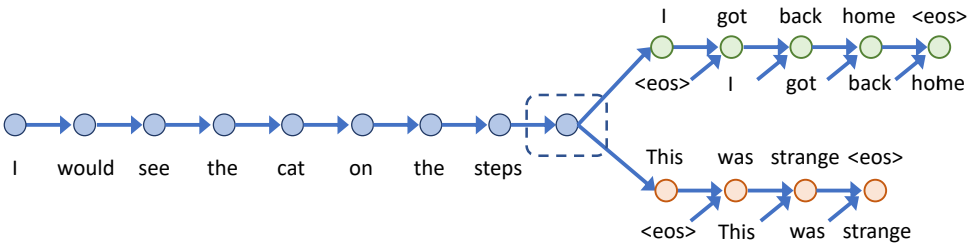

图 4.9 Skip-Thought 模型结构图[22]

在编码器方面，Skip-Thought 模型使用一个 GRU 网络编码输入 s^i，在每个时刻 t 生成表示 $\boldsymbol{h}_t^i$:

$$\boldsymbol{h}_1^i, \ldots, \boldsymbol{h}_N^i = \text{Encoder}(\boldsymbol{x}_1^i, \ldots, \boldsymbol{x}_N^i), t \in [1, T] \tag{4.19}$$

其中 $\boldsymbol{x}_1^i, \ldots, \boldsymbol{x}_N^i$ 是单词 $w_1^t, \ldots, w_N^t$ 的独热表示。在网络中所使用的 GRU 单元的结构计算公式如下所示:

$$\begin{aligned} \boldsymbol{h}_t &= (1 - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \tilde{\boldsymbol{h}}_t \\ \boldsymbol{z}_t &= \sigma\left(\boldsymbol{W}_z \boldsymbol{x}_t + \boldsymbol{U}_z \boldsymbol{h}_{t-1} + \boldsymbol{b}_z\right) \\ \tilde{\boldsymbol{h}}_t &= \tanh\left(\boldsymbol{W}_h \boldsymbol{x}_t + \boldsymbol{r}_t \odot \left(\boldsymbol{U}_h \boldsymbol{h}_{t-1}\right) + \boldsymbol{b}_h\right) \\ \boldsymbol{r}_t &= \sigma\left(\boldsymbol{W}_r \boldsymbol{x}_t + \boldsymbol{U}_r \boldsymbol{h}_{t-1} + \boldsymbol{b}_r\right) \end{aligned} \tag{4.20}$$

解码器的结构对 GRU 进行了部分修改，取编码器在最后一个时刻的输出 $\boldsymbol{h}_N^i$ 作为输入句子的表示 $\boldsymbol{h}_t$，加入到网络输入中。解码器在 t 时刻进行的迭代运算如下：

$$\begin{aligned}\boldsymbol{h}_t &= (1-\boldsymbol{z}_t)\odot\boldsymbol{h}_{t-1}+\boldsymbol{z}_t\odot\tilde{\boldsymbol{h}}_t\\ \boldsymbol{z}_t &= \sigma\left(\boldsymbol{W}_z\boldsymbol{y}_{t-1}+\boldsymbol{U}_z\boldsymbol{h}_{t-1}+\boldsymbol{C}_z\boldsymbol{h}^i\right)\\ \tilde{\boldsymbol{h}}_t &= \tanh\left(\boldsymbol{W}_h\boldsymbol{y}_{t-1}+\boldsymbol{r}_t\odot(\boldsymbol{U}_h\boldsymbol{h}_{t-1})+\boldsymbol{C}_h\boldsymbol{h}^i\right)\\ \boldsymbol{r}_t &= \sigma\left(\boldsymbol{W}_r\boldsymbol{y}_{t-1}+\boldsymbol{U}_r\boldsymbol{h}_{t-1}+\boldsymbol{C}_r\boldsymbol{h}^i\right)\end{aligned} \tag{4.21}$$

其中，修改的 GRU 单元接受三项输入，分别是 $\boldsymbol{h}_{t-1}$，即上一时刻的输出状态；$\boldsymbol{y}_{t-1}$，即上一时刻输出的单词对应的词表示，在自回归解码器中作为下一时刻的输入；$\boldsymbol{h}^i$，即输入句子 s_i 的表示向量。GRU 单元的输出是 $\boldsymbol{h}_t$，即在时刻 t 对输出词语的预测。

给定输出状态后，解码器在每个时刻预测 s^{i-1} 和 s^{i+1} 在当前位置的单词。以解码 s^{i+1} 句子的解码器为例，给定输出表示 $\boldsymbol{h}_t^{i+1}$，对输出词语 w_t^{i+1} 的预测概率满足：

$$P(w_t^{i+1}|w_{<t}^{i+1},\boldsymbol{h}^i)\propto exp(\boldsymbol{v}_{w_t^{i+1}}\boldsymbol{h}_t^{i+1}) \tag{4.22}$$

其中 $\boldsymbol{v}_{w_t^{i+1}}$ 是单词 w_{i+1}^t 的词向量，$\boldsymbol{h}_t^{i+1}$ 是负责编码 s^{i+1} 的编码器在 t 时刻的输出。

模型的训练目标和语言模型相同，即句子 s^{i-1} 和 s^{i+1} 的预测概率最大化：

$$\mathcal{L}(s^i,s^{i+1},s^{i-1})=\sum_t\log P(w_t^{i+1}|w_{<t}^{i+1},\boldsymbol{h}^i)+\sum_t\log P(w_t^{i-1}|w_{<t}^{i-1},\boldsymbol{h}^i) \tag{4.23}$$

通过训练后，我们取最终得到的模型编码器作为句子的表示模型。这个编码器可以为每个句子生成定长向量，且这个向量具有反映句子之间上下文相关关系的性质。

2. Sent2Vec 句子表示模型

为了捕捉句子的语义，需要通过整个句子的全局信息，学习句子中不同词语的关联关系。针对此问题，Sent2Vec 模型[23] 将 CBoW 模型的基于上下文窗口的学习机制扩展到整个句子的范围上，引入词级别的 n 元语法（n-gram）[①]特征提升句子中单词顺序的学习能力，从而更好地捕获上下文语义。在 Sent2Vec 中，模型将句子中所有单词和所有 n 元语法单元的表示向量均值作为句子的表示：

$$\boldsymbol{v}_s=\frac{1}{|R(s)|}\sum_{w\in R(s)}\boldsymbol{v}_w \tag{4.24}$$

① n 元语法详细内容参考本书第**??**节中相关内容

其中，$\boldsymbol{v}_s$ 是句子 s 的表示，$R(s)$ 是句子中全部单词及全部 n 元语法单元集合，$\boldsymbol{v}_w$ 是 $R(s)$ 的元素 w 的上下文词表示，w 可能是一个单词或一个 n 元语法单元。因此，Sent2Vec 词表和对应的嵌入矩阵同时包含单词和 n 元语法单元。

Sent2Vec 的训练目标和 CBoW 类似，通过优化中心词和上下文的相似性量度对文本向量进行自监督训练。具体而言，模型最大化中心词表示和除去该词后其余上下文表示的相似度。同时，Sent2Vec 也采用了负采样的技术，以降低计算成本。对于句子 s 和其中的单词 w，负采样词集合 $N(w)$ 在除 w 外的词表上通过多项式分布采样得到，其中, 记 f_w 为单词 w 的原始词频，单词 w 的采样概率为 $q_n(w) = \frac{\sqrt{f_w}}{\sum_{\mathrm{w}\in\mathcal{V}}\sqrt{f_{\mathrm{w}}}}$. Sent2Vec 的损失函数如下所示：

$$\mathcal{L}(w,s) = \ell(\boldsymbol{u}_w^T \boldsymbol{v}_{s\setminus\{w\}}) + \sum_{w'\in N(w)} \ell(-\boldsymbol{u}_{w'}^T \boldsymbol{v}_{s\setminus\{w\}}) \tag{4.25}$$

其中，$\ell(x) = \log(1+e^{-x})$ 是 Logistic 函数，$\boldsymbol{u}_w$ 是单词 w 的中心词表示，$\boldsymbol{v}_{s\setminus\{w\}}$ 是单词 w 上下文的表示，w' 是负采样词，来自负采样词集 $N(w)$.

在整个语料集上训练时，为了避免模型对高频词的倾向性，采用下采样（subsampling）的方式使模型对单词词频脱敏。对于每个形如 (w,s) 的训练样本，以 $1-q_p(w)$ 的概率丢弃这个样本，其中 $q_p(w) = \min\{1, \sqrt{t/f_w} + t/f_w\}$。Sent2Vec 在整个语料库上训练的损失函数为：

$$\mathcal{L}(\mathcal{D}) = \sum_{s\in\mathcal{D}} \sum_{w\in s} k(w,s)\mathcal{L}(w,s) \tag{4.26}$$

其中 $k(w,s) \in \{0,1\}$ 是概率为 $q_p(w)$ 的伯努利分布在样本 (w,s) 上的取样结果。

4.3.3 篇章分布式表示

在自然语言处理和信息检索领域，部分任务会要求模型学习并表示文档级别的特征，如文档检索、文档去重、文档级情感分析、主题识别等任务。相对一般的自然语言处理任务，这类任务不需要模型精确地捕获细粒度的词句信息，但需要模型建模文档的主题、包含的关键词等信息。编码这些信息成为了文档分布式表示的关键点。本节中将重点介绍词频-逆文档频率（TF-IDF）和 fastText 两种篇章分布式表示方法。

1. 词频-逆文档频率篇章表示方法

词频-逆文档频率（TF-IDF）用来评估在特定文档中词的重要程度，其基本假设是文档中词重要程度随其在文档中出现的频率增加，同时也会随其在整个语料库中出现的频率而下降。对于该假设我们可以从以下角度进行理解，如果一个词在特定文档的出现频率高，则说明这个词与该文档的主题具有比较强的相关关系，因此该词相对于该文档的重要性应该较高。但是，如果一个词语在整个文档集合很多文档上都出现了，那么说明该词是常见词语，其区分性不好，因此其重要程度应该较低。在信息检索领域，TF-IDF 常被用于衡量用户查询和文档之间的相似性。

TF-IDF 方法包括词汇频率（Term Frequency，TF）和逆文档频率（IDF，Inverse Document Frequency）两个部分。具体地，词汇频率主要用来衡量词汇在特定文档中的重要程度，建模文档中的关键词，而逆向文件频率用来衡量词汇在普遍情况下的常见性，用于去除常见词对文档关键词建模的影响。对于文档 d 中的词项 t，$\mathrm{TF}(t,d)$ 表示词项在文档中的出现频率，具体计算方法如下所示：

$$\mathrm{TF}(t,d) = \frac{\mathrm{COUNT}(t,d)}{\sum_{t' \in \mathcal{V}} \mathrm{COUNT}(t',d)} \tag{4.27}$$

其中，$\mathrm{COUNT}(t,d)$ 是文档 d 中的词项 t 的出现次数，分母的和式表示文档中的总词数。

对于词项 t 的逆向文件频率 $\mathrm{IDF}(t)$，计算包含该词项的文档占全体文档的比例，$\mathrm{IDF}(t)$ 与之呈对数反相关关系：

$$\mathrm{IDF}(t) = \log \frac{|\mathcal{D}|}{\sum_{d \in \mathcal{D}} \mathbf{1}^{t \in d}} \tag{4.28}$$

其中，$|\mathcal{D}|$ 表示文档总数，分母的和式表示包含词项 t 的文档数。在计算未登录词项的逆向文件频率时，为了防止分母为零，通常采用平滑化方法，用 $\sum_{d \in \mathcal{D}} \mathbf{1}^{t \in d} + 1$ 代替 $\sum_{d \in \mathcal{D}} \mathbf{1}^{t \in d}$ 作为分母。

词项 t 在文档 d 中的 TF-IDF 表示为词汇频率和逆向文件频率的乘积：

$$\text{TF-IDF}(t,d) = \mathrm{TF}(t,d)\mathrm{IDF}(t) \tag{4.29}$$

文档的表示向量由词表中每个词项在文档中的 TF-IDF 值构成，每个维度对应一个词项。这种表示向量的特点是对关键词信息的反映。对于每个文档，具有较高的 TF-IDF 的词项是在整个文档集合中出现频率较低，但在本文档中出现频率较高的词项。因此，基于 TF-IDF 的表示向量筛选文档中的高频词，过滤掉其中的常见词，保留反映文档主题、主要内容的关键词。在信息检索系统中，对于用户输入的检索关键词，可以容易地通过比较文档中词项的 TF-IDF，来返回与检索词相关性高的文档。

2. fastText 篇章表示模型

fastText 模型[24, 25] 旨在高效训练文本表示模型，因其良好的性能和效率，而被广泛地使用在文本分类任务上。在构建词级别表示时，fastText 会利用字符 n-gram 特征，以更好地表示罕见词和未登录词。具体地，fastText 在单词的开头和末尾添加表示前缀和后缀的字符“<”和“>”，然后通过滑动窗口的方式获得该词固定长度的所有子词。

例如：单词“where”长度为 3 的子词包括：“<wh”，“whe”，“her”，“ere”，“re>”

基于对训练语料的统计，fastText 为全体单词长度不小于 n 的子词建立词表和对应的子词表示矩阵，并使用这些子词辅助构建词表示。例如，n=3 时，使用单词“where”的长度为 3 到 7[①]的全体子词（包括该词自身）辅助构建该词的表示。

① 单词前后增加了“<”和“>”符号，因此单词 where 的长度为 7

fastText 直接使用子词向量的和作为对应单词的词向量：

$$\boldsymbol{v}_w = \sum_{s \in \mathbb{V}_s(w)} \boldsymbol{z}_s \tag{4.30}$$

其中，$\boldsymbol{v}_w$ 是单词 w 的词向量，$\mathbb{V}_s(w)$ 是子词词表中 w 的全体子词，$\boldsymbol{z}_s$ 是子词 s 的表示向量。

fastText 通常使用 Skip-Gram 模型的训练方式得到预训练的词级别表示，其中中心词的向量使用形如公式4.30的方式构建，而其余部分和 Skip-Gram 模型保持一致。在预训练中，fastText 采用 4.3.1 节所述的层次 softmax 方法提升训练效率。

在构建文档表示时，fastText 首先基于上述方式计算每个词的表示向量，再将其进行平均，得到句表示向量。模型结构如图4.10所示，在将 fastText 句向量应用于文本分类任务时，通常以对数概率作为优化目标：

$$\mathcal{L}(x = w_1, \ldots, w_N, y) = -y \cdot \log(softmax(\boldsymbol{W} \cdot \boldsymbol{v})) \tag{4.31}$$

$$\boldsymbol{v} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{v}_i \tag{4.32}$$

其中，$\boldsymbol{v}_i$ 是单词 w_i 的词向量，$\boldsymbol{v}$ 是文档的表示向量，$\boldsymbol{W}$ 是可训练的线性映射层，将样本表示映射为预测分布。

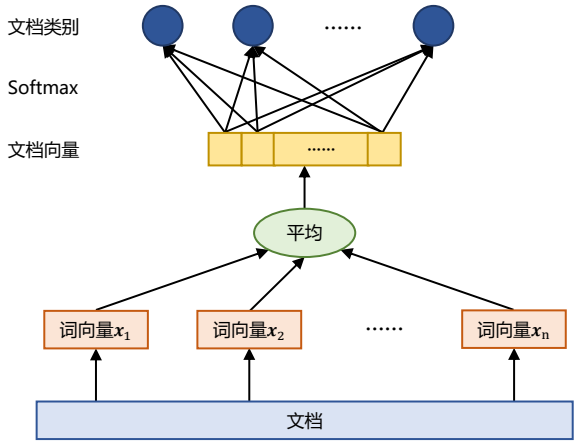

图 4.10　fastText 模型结构图

4.4 词义消歧

词义消歧（Word Sense Disambiguation, WSD）是指确定一个多义词在给定的上下文中具体含义。根据本章第4.1.1节词汇语义学的介绍,我们可以知道语言中一词多义现象十分普遍。例如,“水分”既可以表示物体内所含的水,也可以表示某些情况中夹杂的不真实的成分,可以使用“水分[1]”和“水分[2]”分别表示其两个含义，以如下句子为例：

(1) 葡萄糖液可用来供给水分。单词义项：水分[1]

(2) 这个报导有些水分，需要核实。单词义项：水分[2]

词义消歧任务核心就是根据词语所处句子或者篇章，确定该词在当前环境下的确切含义。上例中句子 (1) 和句子 (2) 中的“水分”分别对应两个不同的义项。该任务对于机器翻译、语义理解、对话系统、阅读理解等任务具有十分重要的作用。本节将介绍基于目标词上下文、基于词义释义匹配以及基于词义知识增强预训练等三类词义消歧方法。

4.4.1 基于目标词上下文的词义消歧方法

对于待消歧的目标词，词义消歧方法通常采用有监督分类方法，将词语的每个词义项作为候选词义，通过估计待消歧词义的概率分布从而完成目标词的词义消歧。基于目标词上下文的词义消歧方法利用待消歧目标词的上下文进行训练，预测上下文中目标词属于每个候选词义的条件概率。自然语言处理中常用的统计机器学习方法和深度学习算法，均可用于构建基于目标词上下文的词义消歧方法。本节以基于朴素贝叶斯分类器、上下文向量表示的词义消歧方法为例，介绍基于统计机器学习及深度学习的词义消歧方法。

1. 基于朴素贝叶斯分类器的消歧方法

使用 w 表示待消歧的目标词，c 表示目标词所处的句子，$\{s_i\}_{i=1}^N$ 为目标词的候选词义集合。由于待消解的目标词词义仅与其所处的上下文语境有关，因此可以通过估计条件概率 $P(s_i|c)$ 来预测目标词 w 的词义[26]。

给定句子 $c = \{w_k\}_{k=1}^K$，可以将条件概率 $P(s_i|c)$ 通过贝叶斯公式转换为：

$$P(s_i|c) = \frac{P(c|s_i)P(s_i)}{P(c)} \propto P(c|s_i)P(s_i) \tag{4.33}$$

其中，$P(c|s_i)$ 的计算方式与语言模型类似。随着上下文长度的增加，上下文 c 的数量指数级增长，该概率值难以估计。因此，需要引入单词独立假设，近似地将概率估算为上下文中每个单词的独立出现概率：

$$P(c|s_i) = \prod_{k=1}^{K} P(w_k|s_i) \tag{4.34}$$

通过公式4.33和公式4.34，词义分类可以根据如下公式选择最大条件概率的词义：

$$\hat{s} = \arg\max_{s_i} P(s_i|c) = \arg\max_{s_i} P(s_i) \prod_{k=1}^{K} P(w_k|s_i) \tag{4.35}$$

$P(s_i)$ 和 $P(w_k|s_i)$ 可以通过训练语料利用最大似然估计得到：

$$P(w_k|s_i) = \frac{\text{COUNT}(w_k, s_i)}{\text{COUNT}(s_i)} \tag{4.36}$$

$$P(s_i) = \frac{\text{COUNT}(s_i)}{\text{COUNT}(w)} \tag{4.37}$$

其中，$\text{COUNT}(w_k, s_i)$ 是训练语料中目标词 w 以语义 s_i 在上下文中出现的次数；$\text{COUNT}(s_i)$ 是训练语料中语义 s_i 出现的总次数；$\text{COUNT}(w)$ 是训练语料中目标词 w 出现的总次数。

在实际算法实现中，为了提升计算精度，概率值通常采用对数形式参与计算。具体的算法如4.3所示：

代码 4.3: 基于朴素贝叶斯分类器的词义消歧算法

输入: 训练数据 D，语句 c，目标词 w，候选语义 $\{s_i\}_{i=1}^{N}$
输出: 预测的词义 $\hat{s}$
for $i = 1$ to N **do**
  for $k = 1$ to K **do**
    $P(w_k|s_i) = \text{COUNT}(w_k, s_i)$ / $\text{COUNT}(s_i)$ // 根据训练数据计算单词出现概率
  end
  $P(s_i) = \text{COUNT}(s_i)$ / $\text{COUNT}(w)$ // 根据训练数据计算语义出现概率
end
$\hat{s} = \arg\max_{s_i} log\ P(s_i) \sum_{k=1}^{K} \log\ P(w_k|s_i)$ // 选择出现概率最大的词义
return $\hat{s}$

2. 基于上下文向量表示的消歧方法

本章第 4.3 节分布式表示中介绍了句子和短语分布式向量表示算法，可以看到深度神经网络算法可以很好地对句子和短语的语义进行表示，并在各种自然语言处理任务中都得到了广泛的应用。词义消歧任务的核心就是根据上下文信息判断当前单词的词义，因此，也可以利用目标词上下文的分布式表示，建模目标词上下文语义，并基于上下文向量表示构建词义消歧算法。

基于上下文向量表示的最近邻方法[27, 28] 将词义消歧任务形式化为词义表示和上下文表示的相似度学习问题。如图4.11所示，该方法的主要过程是根据词义在训练集样本中出现的上下文，学习词义的表示。根据语料库训练目标词的上下文表示和候选词义表示的相似度计算模型。在应用

时，将每个候选词义的表示和目标词的上下文表示进行比较，选择相似度最高的词义。针对未在词义消歧语料库中出现的词义，可以进一步使用相似的语义集合确定其词义表示。

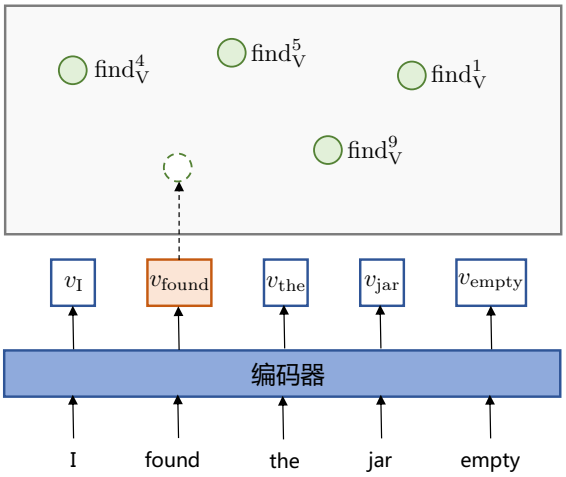

图 4.11 基于上下文向量表示的最近邻模型结构图[27]

在词义编码部分，首先考虑在词义消歧语料库中存在标注的语义。对于每一个标注词义，在训练集中抽取全体包含该词义标注的样本。随后，通过预训练上下文表示模型，计算词义对应的目标词在样本上下文中的表示。最后，以目标词表示的平均值作为词义的表示。具体地，对于词义 s，若该词义在词义消歧语料库出现过，则根据如下公式计算该词义表示 $\boldsymbol{v}$：

$$\boldsymbol{v_0}, \boldsymbol{v_1}, \ldots, \boldsymbol{v_w}, \ldots, \boldsymbol{v_n} = \text{Encoder}(c = c_0, c_1, \ldots, w, \ldots, c_n) \tag{4.38}$$

$$\boldsymbol{v} = \frac{1}{|C(s)|} \sum_{c \in C(s)} \boldsymbol{v_w} \tag{4.39}$$

其中，$C(s)$ 为全体标记词义为 s 的样本集合，Encoder 代表使用预训练语言模型初始化的编码器，如 ELMo、BERT 等。在 BERT 模型中，对于分解成多个标识位的单词，采用每个标识位输出的平均值作为单词的表示。w 是词义在样本中对应的目标词，在不同的样本中，目标词不必相同。$\boldsymbol{v_w}$ 是目标词 w 在上下文 c 中的表示，$\boldsymbol{v}$ 是词义 s 的上下文表示。

针对未在词义消歧语料库中出现的词义，可以采用 LMMS[29] 方法，利用 WordNet 中标注的同义词（synset）、上位词（hypernym）和词性标注（lexname）等语义关系信息，寻找与目标词义相似或相关的词义，再以这些词义表示的平均值作为该词义的表示。具体地，以同义词关系为例，

对于待确定表示的词义 s，记 $S(s)$ 为 s 的同义语义集合。若 $S(s)$ 不是空集，s 的语义表示为 $S(s)$ 中同义语义的平均表示：

$$\boldsymbol{v} = \frac{1}{|S(s)|} \sum_{\mathrm{s} \in S(s)} \boldsymbol{v}_{\mathrm{s}}, \text{if } |S(s)| > 0 \tag{4.40}$$

当同义语义缺失时，可依次使用相同上位的语义或相同词性的语义作为近义语义集合，利用相似的方式计算目标语义的表示，具体计算公式如下所示：

$$\boldsymbol{v} = \frac{1}{|H(s)|} \sum_{\mathrm{s} \in H(s)} \boldsymbol{v}_{\mathrm{s}}, \text{if } |H(s)| > 0 \tag{4.41}$$

$$\boldsymbol{v} = \frac{1}{|L(s)|} \sum_{\mathrm{s} \in L(s)} \boldsymbol{v}_{\mathrm{s}}, \text{if } |L(s)| > 0 \tag{4.42}$$

其中 $H(s)$ 为 s 的同上位语义集合，包含与 s 属于同类概念的语义。$L(s)$ 为 s 的同词性语义集合，包含与 s 词性相同的全体语义。在结合语义关系之后，可以全面覆盖 WordNet 中的词义，从而解决候选词义未在训练集出现的问题。

在构建了所有词义的向量表示后，对于每一条输入的待进行词义消歧的样本，首先基于语言模型计算目标词的上下文表示，在此基础上，计算上下文表示与全体候选词义表示的点积相似度，选择相似度最大的语义做为分类结果，具体计算公式如下所示：

$$\hat{s} = \arg\max_{s} \left(\boldsymbol{v}_{\boldsymbol{w}} \cdot \boldsymbol{v}(s)\right) \tag{4.43}$$

其中，$\boldsymbol{v}_{\boldsymbol{w}}$ 为目标词的上下文表示，和词义表示映射到相同维度 $\boldsymbol{v}(s)$ 为候选词义 s 的表示。

4.4.2 基于词义释义匹配的词义消歧方法

以知网（HowNet）[3]、WordNet[7] 等为代表的词汇知识资源中不仅包含了词义之间的关系，还包含了词义的解释信息。

例如：WordNet 3.1 中对“table”给出了如下词义解释：

table^1: a set of data arranged in rows and columns

table^2: a piece of furniture having a smooth flat top that is usually supported by one or more vertical legs

table^3: a piece of furniture with tableware for a meal laid out on it

table^4: flat tableland with steep edges

table^5: a company of people assembled at a table for a meal or game

table^6: food or meals in general

这些释义与目标词上下文之间存在着非常强的联系。比如，table^1 所对应的“表格”含义，其上下文

更多的对应的设计、制作、数据等词汇。而 table[2] 所对应的“桌子”含义，其上下文更多的对应的椅子、沙发等词汇。因此，也可以将词义消歧问题转化为目标词上下文和词义释义之间的语义匹配问题。对于待消歧的目标词 w 所在的上下文句子 c，以及候选词义 s，构建相似度度量函数，建模目标词上下文和候选词义的匹配度。在应用时，根据目标词和每个候选语义的相似度得分，来确定目标词义的预测分布 $\phi(w|c,s)$。本节将以基于特征式和交互式匹配的两类消歧方法为例，介绍基于词义释义匹配的词义消歧方法。

1. 基于特征式匹配的消歧方法

BEM 模型[30] 通过分布式向量表示匹配方式学习目标词上下文和词义释义的相似性。BEM 模型主要包含上下文编码器和词义编码器两个组成部分。上下文编码器 T_c 对输入的目标词及其上下文进行编码，计算目标词上下文的分布式表示。词义编码器 T_g 对输入的词义释义文本进行编码，将输入词义和上下文表示在同一表示空间内。通过建立上下文语义表示和候选词义表示的相似度计算模型，来完成词义消歧任务。BEM 模型结构如图4.12所示。

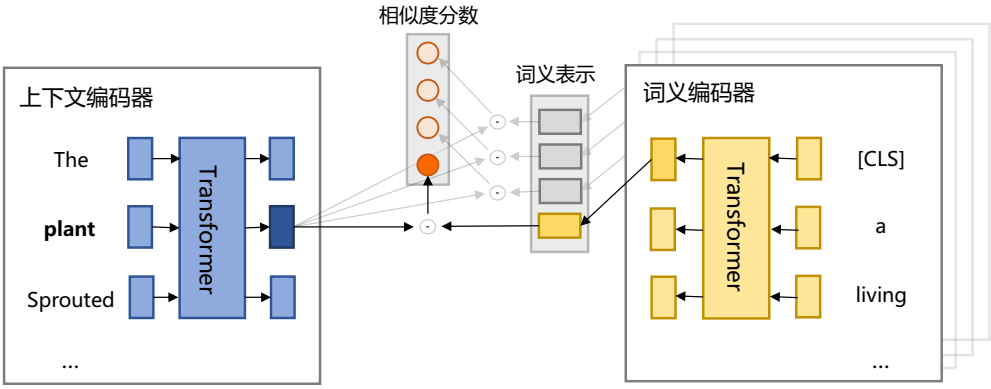

图 4.12　BEM 模型结构图[30]

BEM 的模型结构的上下文编码器 T_c 和词义编码器 T_g 都采用基于 BERT 的架构，参数使用语言模型进行预训练。针对目标词上下文表示的计算,使用 w 表示待消歧的目标词,$c = c_0, c_1, \ldots, w, \ldots, c_n$ 表示目标词所处的上下文，上下文编码器 T_c 将序列中的每一个单词构建对应的上下文表示，并取目标词位置的输出为目标词的上下文表示，具体计算公式如下所示：

$$\boldsymbol{v_{cls}}, \boldsymbol{v_0}, \boldsymbol{v_1}, \ldots, \boldsymbol{v_w}, \ldots, \boldsymbol{v_n}, \boldsymbol{v_{sep}} = T_c([CLS], c_0, c_1, \ldots, w, \ldots, c_n, [SEP]) \tag{4.44}$$

其中 $\boldsymbol{v_w}$ 是目标词 w 在句子中的上下文表示。对于分解成多个标识位的单词，采用每个标识位输出的平均值作为该单词的表示。

针对候选词义表示的计算，候选词义 s 的词义释义为 $g_s = g_0, g_1, \ldots, g_m$，在词义释义序列的首尾添加 [CLS] 及 [SEP] 标识，输入词义编码器 T_g，取 [CLS] 位置的输出作为词义的表示。计算过程如下所示：

$$\boldsymbol{v}_{\boldsymbol{cls}}^{\boldsymbol{g}}, \boldsymbol{v}_{\boldsymbol{0}}^{\boldsymbol{g}}, \boldsymbol{v}_{\boldsymbol{1}}^{\boldsymbol{g}}, \ldots, \boldsymbol{v}_{\boldsymbol{m}}^{\boldsymbol{g}}, \boldsymbol{v}_{\boldsymbol{sep}}^{\boldsymbol{g}} = T_g([CLS], g_0, g_1, \ldots, g_m, [SEP]) \tag{4.45}$$

其中，记 $\boldsymbol{v}_{\boldsymbol{s}} = \boldsymbol{v}_{\boldsymbol{cls}}^{\boldsymbol{g}}$ 为候选词义 s 的表示。

对于上下文 c 中待消歧的目标词 w，以及候选词义 s，它们的相似度由如下公式计算得到：

$$\phi(w|c, s) = \boldsymbol{v}_{\boldsymbol{w}} \cdot \boldsymbol{v}_{\boldsymbol{s}} \tag{4.46}$$

其中 $\boldsymbol{v}_{\boldsymbol{w}}$ 是目标词的表示向量，$\boldsymbol{v}_{\boldsymbol{s}}$ 是词义 s 的表示向量。

在模型训练过程中，对于待消歧的目标词 w，取该目标词在句子中的表示与全体候选词义的表示进行相似度计算，以相似度作为预测词义的对数概率分布，优化交叉熵损失函数，具体计算公式如下：

$$\mathcal{L}(w; c) = -\phi(w|c, s^+) + \log \sum exp(\phi(w|c, s^-)) \tag{4.47}$$

其中，s^+ 表示正确的候选词义，s^- 表示其余的候选词义。

在使用模型进行词义消歧时，与目标词具有最大相似度的词义将被预测为目标词的词义，即：

$$\hat{s} = \arg\max_{s} \ \phi(w|c, s) \tag{4.48}$$

2. 基于交互式匹配的消歧方法

基于深度神经网络的交互式语义匹配算法在自然语言处理任务上取得了不错的效果。在交互式匹配中，待判断匹配的文本对以特定的方式拼接在一起输入模型，然后以与单文本相同的方式进行处理。交互式匹配的优点是只使用一个编码器进行匹配任务，大大减小了训练参数的规模。此外，交互式匹配可以充分利用词粒度的信息，参考输入的一对文本中的每个单词，进行充分的比较，从而实现更好的学习效果。交互式匹配方法也可以应用于上下文和词义释义的相似度学习中。

GlossBERT[31] 使用交互式匹配方法，通过对预训练模型 BERT 进行微调，实现上下文和词义释义的相似度计算。GlossBERT 以 BERT 双句分类的方式，将目标词所处的上下文句子和词义释义组合为输入，以语义是否匹配作为二分类标签，构造分类模型的微调样本，通过这些样本进行模型的微调。模型通过微调后，对于待消歧的目标词和候选词义，将目标词上下文和每一个候选词义组合成输入，通过模型计算语义匹配的置信度，根据置信度选取预测词义。

对于微调样本的构造，表4.6通过一个示例样本展示了 GlossBERT 微调样本的格式。该句以“research”作为消歧目标词时，目标词具有“research%1:04:00::”等 4 个词义。GlossBERT 按照输入模板将上下文句子和各个词义的释义分别拼接为 BERT 的输入形式，并在输入词序列中使用双引号

(”)、冒号（:）等标点符号标出目标词的位置。对于和目标词匹配的词义，GlossBERT 为微调样本赋予“Yes”类标签，否则赋予“No”类标签。

表 4.6 GlossBERT 的微调样本构造，目标词用斜体表示。

原句：Your *research* stopped when a convenient assertion could be made .		
微调样本	标签	原始词义
[CLS] Your ”research” ... [SEP] research: systematic investigation to ... [SEP]	Yes	research%1:04:00::
[CLS] Your ”research” ... [SEP] research: a search for knowledge [SEP]	No	research%1:09:00::
[CLS] Your ”research” ... [SEP] research: inquire into [SEP]	No	research%2:31:00::
[CLS] Your ”research” ... [SEP] research: attempt to find out in a ... [SEP]	No	research%2:32:00::

整个过程可以形式化地表示为：给定待消歧对目标词 w 及其所处的上下文句子 $c = c_0, c_1, \ldots, w, \ldots, c_n$，对于 w 的每个候选词义 s_i 的释义 $g^i = g_0^i, \ldots, g_m^i$，将句子和词义组合为如下的输入形式：

$$\begin{aligned} x &= f(w, c, s_i) \\ &= [CLS], c_0, c_1, \ldots, ”, w, ”, \ldots, c_n, [SEP], w, :, g_0^i, \ldots, g_m^i, [SEP] \end{aligned} \tag{4.49}$$

其中，[CLS] 和 [SEP] 是 BERT 使用的特殊标识，双引号（”）、冒号（:）是用来标出目标词位置的特殊标识。如果目标词实际词义和词义释义匹配，则标注正例标签，否则标注负例标签。因此，对于样本中每个待消歧的目标词，可以构造 N 个分类样本，其中 N 是目标词候选词义的个数。

GlossBERT 的模型结构如图4.13所示，其中，BERT 分类层包含一个 BERT 编码层和一个线性分类层。如公式4.50所示，GlossBERT 根据训练集中每个样本的每个目标词所构造的分类样本，使用 BERT 编码层在 [CLS] 位置的输出作为分类判据，通过下游的线性分类层进行语义是否匹配的二元分类。

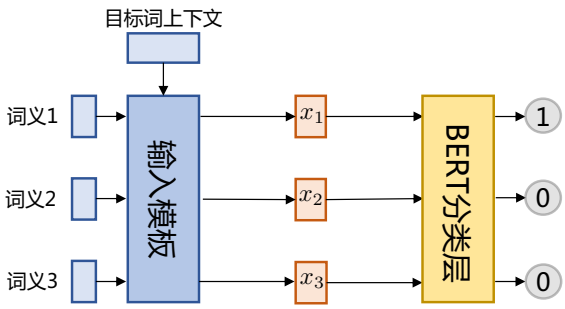

图 4.13 GlossBERT 模型结构图

$$\hat{y} = \boldsymbol{W} \cdot \text{BERT}(x)[0] \tag{4.50}$$

其中 $\boldsymbol{W}$ 为线性分类层的权重。

GlossBERT 根据交叉熵损失函数微调 BERT 编码层及线性输出层的全部权重：

$$\mathcal{L}(x, y) = \text{CrossEntropy}(\hat{y}, y) \tag{4.51}$$

当使用模型进行词义消歧时，将目标词构造对应的分类样本，输入模型对其进行预测。预测正例标签置信度最高的候选词义将被预测为目标词的词义。

$$\hat{s} = \arg\max_{s} \ P(\hat{y}(w, c, s) = 1) \tag{4.52}$$

4.4.3 基于词义知识增强预训练的消歧方法

基于预训练语言模型的方法在词义消歧任务中取得了不错的结果，为了使得预训练语言模型更好地适应词义消歧任务，可以通过设计词义级别的预训练任务，使得预训练模型融合知识库中所包含词义信息。然而，预训练模型需要大规模的有监督数据，才能对模型参数进行有效训练。但是，目前缺乏标注了词义的大规模数据用于支持模型预训练。

SenseBERT 模型[32]，针对缺失语义监督问题，在 BERT 的预训练中添加了一个掩码词义预测任务作为辅助任务。SenseBERT 利用 WordNet 所包含的超义（Supersense）信息作为弱监督信号。WordNet 将所有义项归纳为多个类别，这些类型称之为超义。例如，针对名词有 26 个超义，包括：BODY、LOCATION、PLANT 等。模型在预训练中不但预测掩码位置单词的词形，还预测缺失单词的超义。通过这种方式引入语义级别的监督信息，从而提升预训练模型在词义消歧任务上的效果。

SenseBERT 的模型结构如4.14所示，在输入表示部分增加了语义嵌入模块，用来建构单词的词义信息。在 SenseBERT 中，对于输入序列 $w_1, \ldots, w_n$，每个标识位的输入表示 $\boldsymbol{v_t}$ 由词形嵌入、词义嵌入和位置嵌入的和组成：

$$\boldsymbol{v_t} = \boldsymbol{e_t}(\boldsymbol{W_t} + \boldsymbol{W_s}) + \boldsymbol{p_t}\boldsymbol{W_p} \tag{4.53}$$

其中，$\boldsymbol{e_t} \in \mathbb{R}^{|\mathbb{V}|}$ 是第 t 个位置上标识的独热向量表示，$\boldsymbol{W_t} \in \mathbb{R}^{|\mathbb{V}| \times d_t}$ 是词嵌入矩阵，$\boldsymbol{W_s} \in \mathbb{R}^{|\mathbb{V}| \times d_s}$ 是语义嵌入矩阵，$|\mathbb{V}|$，d_t 和 d_s 分别表示词表大小、词嵌入维度和语义嵌入维度，其中语义嵌入维度为超义类型的个数；$\boldsymbol{p_t} \in \mathbb{R}^N$ 是标识所在位置的独热向量表示，$\boldsymbol{W_p} \in \mathbb{R}^{N \times d_p}$ 是位置嵌入矩阵，N 和 d_p 分别表示位置长度上限和位置嵌入维度。

SenseBERT 使用和 BERT 相同的 Transformer 编码层作为输入的编码结构：

$$\boldsymbol{o_1}, \ldots, \boldsymbol{o_n} = \text{Transformer-Encoder}(\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}) \tag{4.54}$$

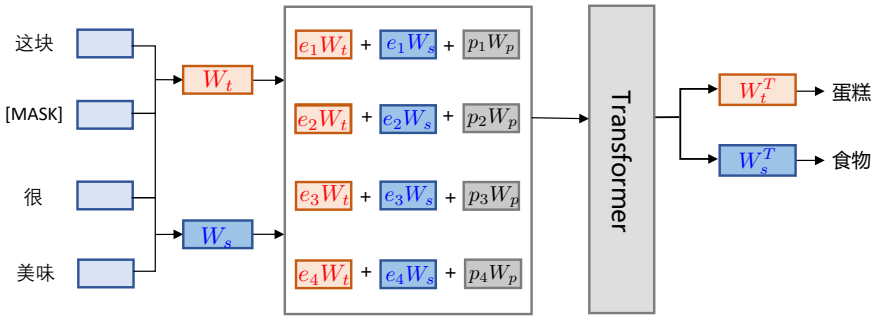

图 4.14 SenseBERT 模型结构图[32]

在预训练任务方面,SenseBERT 包括掩码单词预测和掩码语义预测两个任务。如公式4.55和4.56所示，通过与词嵌入、语义嵌入矩阵的比较，模型计算每一个掩码位置的单词预测分布和语义预测分布，并将其与实际标签比对。

$$P(\hat{w}_t|context) = \text{softmax}(\boldsymbol{W_t}^T \boldsymbol{o_t}) \tag{4.55}$$

$$P(\hat{s}_t|context) = \text{softmax}(\boldsymbol{W_s}^T \boldsymbol{o_t}) \tag{4.56}$$

在掩码单词预测任务上，如公式4.57所示，模型约束单词预测分布和实际单词的交叉熵损失。根据 WordNet 统计实际单词全体词义所属的超义，单词 w 的超义集合记为 $A(w)$，即单词 w 的可取义项，作为掩码语义预测的弱监督信号。

$$\mathcal{L}_{LM}(w_t) = -\log P(\hat{w}_t|context) \tag{4.57}$$

在掩码语义预测的优化中，要求模型预测与超义集合中的语义吻合即可。如公式4.58所示，损失项 $\mathcal{L}_{SLM}^{allowed}(w_t)$ 优化模型预测掩码位置语义 $\hat{s}_t$ 属于可取义项集合 $A(w_i)$ 的概率。考虑到模型在训练过程中可能会出现过拟合特定超义义项的情况，SenseBERT 引入了正则项 $\mathcal{L}_{SLM}^{reg}(w_t)$。如公式4.59所示，此损失项对 $A(w_i)$ 中每个超义义项的模型预测置信度的平均值进行约束，保证模型对全体可取义项都有一定的预测置信度，防止出现模型对单一义项预测置信度过高，预测分布退化的情况。

$$\mathcal{L}_{SLM}^{allowed}(w_t) = -\sum_{s_t \in A(w_t)} \log P(\hat{s}_t|context) \tag{4.58}$$

$$\mathcal{L}_{SLM}^{reg}(w_t) = -\sum_{s_t \in A(w_t)} \frac{1}{|A(w_t)|} \log P(\hat{s}_t|context) \tag{4.59}$$

模型在预训练阶段的整体损失函数为掩码单词预测和掩码语义预测目标的加权求和值:

$$\mathcal{L} = \sum_{w_t=[MASK]} \lambda_1 \mathcal{L}_{LM}(w_t) + \lambda_2 \mathcal{L}_{SLM}^{allowed}(w_t) + \lambda_3 \mathcal{L}_{SLM}^{reg}(w_t) \tag{4.60}$$

由于掩码词义预测只对被掩盖的全词有意义, SenseBERT 在掩码标识为子词时, 只优化掩码单词预测损失, 不训练掩码语义预测任务。为了进一步增强对词义知识的学习, SenseBERT 在 BERT 的原始词表增加了维基百科中的高频词, 使掩码全词的比例更大, 有助于学习低频词义。

4.4.4 词义消歧评价方法

词义消歧评测的的主要指标通常采用机器学习算法评测常用指标, 包括词义消歧的精度 (Precision), 召回率 (Recall), 和 F 值 (F-Score) 得分。在词义消歧任务中的计算方式如下:

$$\text{精度 (P)} = \frac{\text{算法输出的正确标记个数}}{\text{算法输出的全部标记个数}} \times 100\% \tag{4.61}$$

$$\text{召回率 (R)} = \frac{\text{算法输出的正确标记个数}}{\text{测试集合中全部标记个数}} \times 100\% \tag{4.62}$$

$$\text{F 值 (F1)} = \frac{2 \times P \times R}{P + R} \tag{4.63}$$

在部分基于机器学习的方法中, 也会使用精度和覆盖率 (Coverage) 作为词义消歧系统的评测指标[33]。覆盖率的计算方式如4.64所示:

$$\text{覆盖率} = \frac{\text{算法输出的全部标记个数}}{\text{测试集合中全部标记个数}} \times 100\% \tag{4.64}$$

4.4.5 词义消歧语料库

大规模词义消歧标注语料库和评测竞赛自有监督机器学习算法大规模应用于词义消歧开始发展迅速。1997 年以来, 国际计算语言学联合会 (ACL) 的词法研究小组 (SIGLEX) 开始组织关于词义消歧的公共评测任务 SensEval, 发布了词义消歧任务的训练集和测试集, 用于评测词义消歧任务的性能。SIGLEX 同样也支持了 SemEval 研讨会的举办, 在每年的研讨会上发布一部分有挑战性的共享任务, 建立高质量的注释数据集。

词义消歧语料库主要包含两种类型: 义项分类和义项相同判断。义项分类语料库针对目标词语, 构建包含该目标词句子, 并对句子中目标词所属的语义项进行标注。义项相同判断语料库则

针对目标词给出两个包含该词语的句子，并依据在两个句子中目标词的词义是否相同给出分类标签。本节将分别介绍上述两种常用词义消歧系统训练的大规模语料库。

1. 词义消歧义项分类标注语料库

SemCor[34] 是迄今为止最大的手工标注词义的语料库之一。目前的词义消歧算法研究中，大多数有监督方法均使用 SemCor 语料库进行训练。SemCor 是基于 WordNet 词义进行标注的语料库。SemCor 3.0 版本包含 352 个文档和 22 万余条手动语义注释，其原始语料从布朗（Brown）语料库获取，经过筛选后，参考 WordNet 1.4 的词义清单进行词义标记。SemCor 分为三部分，其中 Brown1，Brown2 分别包含 103 个和 83 个文件，标记了所有开放类型的词语；Brownv 包含 166 个文件，只标注了动词。在这些文件中，对于文中的每一个句子或段落，标注其中每个单词的词性，并对名词、动词、形容词和副词标记来自 WordNet 的义项和语义。表4.7展示了 SemCor 的一个标注数据样例，在句中标出了开放域词汇，如“got. . . up”；包含多个词汇的表达，如“on their feet”；以及命名实体，如“Kim”以实体类型进行标注[35]。

表 4.7　SemCor 语料库的示例样本标注

例句：Kim_a got_b slowly_c up_b , the children_d were_e already_f on_g their_g feet_g .			
编号	范围	对应词	词义
a	Kim	Kim	**org**
b	got, up	get_up	get_up_4
c	slowly	slowly	slowly_1
d	children	child	child_1
e	were	be	be_3
f	already	already	already_1
g	on their feet	on_one’s_feet	**no_tag**

OMSTI（One Million Sense-Tagged Instances）[36] 是自动标注的语料库，也常用于词义消歧系统的训练。OMSTI 使用 WordNet 3.0 的词义进行注释，它是通过在大型英汉平行语料库（MultiUN 语料库）上使用基于对齐的词义消歧方法自动构建的。OMSTI 中包含针对 22437 个单词，由 85 万句子组成的 113 万训练样例。虽然 OMSTI 是自动标注的，但其具有比 SemCor 更大的规模，且包含更丰富的歧义情况，很多工作的实验说明将 OMSTI 应用于词义消歧算法训练，可以有效提升算法效果。

WSDEvaL[37] 是统一词义消歧基准评测框架，将不同时期构建的采用不同词义注释构建的评测基准语料统一使用 WordNet 3.0 词义进行注释。WSDEvaL 包含 5 个来自 SensEval 和 SemEval 的测试基准语料，具体如下：

- SensEval-2 使用 WordNet 1.7 进行标注，包含 2282 个词语义标注。
- SensEval-3 Task 1 使用 WordNet 1.7.1，包含来自社论，新闻和小说领域的 3 个文档，包含 1850

个词语义标注。

- SemEval-07 Task 17 使用 WordNet 2.1，包含 455 个词语义标注。
- SemEval-13 Task 12 使用 WordNet 3.0 的标注，包含来自多个领域的 13 个文档和名词上的 1644 个词语义标注。
- SemEval-15 Task 13 使用 WordNet 3.0 的标注，包含生物医学、数学/计算和社会问题题材的 4 个文档和 1022 个词语义标注。

2. 词义消歧义项相同判断标注语料库

WiC（Word in Context）数据集是一个由专家标注的词义消歧数据集，每个样本对同一个目标词给出两个包含该词语的句子，并依据在两个句子中目标词的词义是否相同，给出 T 或 F 的分类标签。表4.8给出了 WiC 数据集中的一些示例样本。在 WiC 的构建过程中，句子主要来自 WordNet 中的例句语料，并使用了一部分来自 VerbNet 和 Wiktionary 的语料资源。在这些句子中，只取具有多种含义、并出现在不同句子中的名词和动词作为目标词。为了筛选高质量的验证集和测试集，标注者要求包含相同目标词的样本不超过 3 个，且样本之间没有重复的上下文句子。在此基础上，尽量保证类别平衡、以及目标词的多样性。表4.9给出了 WiC 数据集的统计数据，包括样本个数、目标词个数和目标词中名词、动词的占比。

表 4.8　WiC 数据集的示例样（目标词用斜体表示，同义/非同义的语句对标记为 T/F）

语句对	标签
There is a lot of trash on the *bed* of the river I keep a glass of water next to my *bed* when I sleep	**F**
Justify the margins The end *justifies* the means	**F**
Air pollution Open a window and let in some *air*	**T**
The expanded *window* will give us time to catch the thieves You have a two-hour *window* of clear weather to finish working on the lawn	**T**

表 4.9　WiC 数据集样本统计

数据划分	样本个数	名词比例	动词比例	目标词数
训练集	5,428	49%	51%	1,256
验证集	638	62%	38%	599
测试集	1,400	59%	41%	1,184

WiC 用准确率作为任务的评价指标。与 SemEval 和 SensEval 评测基准相比，WiC 不依赖于 WordNet 等外部数据库的词义信息，而且作为分类数据集，具有较为简单的任务形式。简洁的任

务形式促进了模型的灵活性和丰富性，而不依赖于外部信息的特质使 WiC 支持低资源场景下的评测，而不强制假设完整训练数据的可用性。因为 WiC 的简洁性、独立性，以及其对模型语义理解能力较高的要求，SuperGLUE 系列基准测试任务收录 WiC 任务作为词义消歧方面的评测基准，用来衡量现代高性能语义理解智能系统的表现。

在 WiC 的基础上，针对词义消歧系统对专用领域词义的建模，WiC-TSV（Word in Context - Target Sense Verification）对 WiC 的语料筛选和任务形式进行了改进，形成了新的跨越多个领域的词义消歧评测基准。与 WiC 不同，WiC-TSV 中的每个样本仅包含一个句子，其中标出待消歧的目标词。同时，样本包含该单词的一个预期词义，根据目标词在上下文中的实际词义是否和给出的词义相符，标记 T 或 F 的分类标签。另外，样本中还会给出单词词义的上位词，保证预期词义和实际词义在领域上是接近的。表4.10给出了 WiC 数据集中的一些示例样本。

表 4.10 WiC-TSV 数据集的示例样本（目标词用斜体表示，语义匹配/不匹配的样本标记为 T/F）

数据划分	语句	预期词义	上位词	标签
WNT/WKT	*Smoking* is permitted .	the act of smoking tobacco or other substances	breathing, respiration, ventilation	**T**
	All that work went down the *sewer* .	someone who sews	needleworker	**F**
CTL	We started the evening with *Bellini* , made with fresh , Niagara peaches .	A Bellini cocktail is a mixture of Prosecco sparkling wine and peach purée	cocktail	**T**
	After a morning 's work I went off to see the *Bellini* retrospective at the Quirinale .	A Bellini cocktail is a mixture of Prosecco sparkling wine and peach purée	cocktail	**F**
MSH	Italy now reports the second highest number of *corona* cases worldwide .	A viral disorder characterized by high fever, ... and other symptoms of a viral pneumonia.	viral pneumonia; coronavirus infection	**T**
	Corona Labs is happy to announce the general availability of the public beta of Android 64-bit Corona builds .	A viral disorder characterized by high fever, ... and other symptoms of a viral pneumonia.	viral pneumonia; coronavirus infection	**F**
CPS	pandas is an open source data analysis and manipulation tool built on top of the *Python* programming language .	Python is an interpreted, high-level, general-purpose programming language	programming language	**T**
	The present paper compares the recently studied *pythons* with those examined 20 years ago , and uses the combined dataset to assess the ecological sustainability .	Python is an interpreted, high-level, general-purpose programming language	programming language	**F**

WiC-TSV 的原始语料同样来自 WordNet 和 Wiktionary，构造的训练和验证集包含通用域的样

本，而测试集包含通用域和专用域的样本。具体而言，WNT/WKT 测试集包含通用域的样本，而 Cocktails（CTL），Medical Subjects（MSH）和 Computer Science（CPS）测试集分别包括酒饮、医疗和计算机科学领域的测试样本。

4.5 语义角色标注

语义角色标注（Semantic Role Labeling, SRL）是一种浅层语义分析技术，目标是分析句子的谓词-论元结构，揭示句子中概念范畴之间的语义关系。语义角色标注的主要语言学理论来源于题元理论 (Thematic Theory)、格语法（Case Grammar）以及配价理论（Valency Theory）句子语义理论等。题元理论认为句子以谓语为中心，谓语决定了句子的基本结构。**论元**（Argument）是谓语所涉及的对象，担任了施事、客体、受事、地点或命题等不同的题元角色。语义角色标注任务核心是识别句子中谓语的论元，并确定论元的题元角色。

例如：$[\text{美国波音公司}]_{A0}[\text{正在}]_{AM-TMP}[\text{制造}]_{V}[\text{民用飞机}]_{A1}$。
“制造”为谓词（V），代表了一个事件的核心行为；“美国波音公司”和“民用飞机”为动作的施事者（A0）和受事者（A1）。

在语义角色标注任务研究早期，相关算法往往依赖句子的句法结构。近年来，得益于机器学习和深度学习方法的不断发展，不依赖句法结构信息的语义角色标注方法研究也逐渐兴起。语义角色标注算法虽然有很多类型，但是其基本基本流程都主要由论元识别和论元标注组成。基于句法分析的语义角色标注算法还需要先对句子进行句法分析。论元识别的目标是从句子识别所有由连续几个单词组成的论元。由于如果将句子中所有的连续单词片段都为论元候选，其数量会过于庞大，因此早期的方法在进行论元识别前，通常还会引入基于规则的候选论元过滤方法，利用句法分析结果构造启发式规则对候选项进行大幅度删减。论元标注则是对论元和谓词之间的关系类型进行标注。论元识别和论元分类通常采用有监督机器学习算法，将上述任务转换为分类问题。两个任务之间可以采用流水线结构，也可以采用联合学习的方法。

本节中将介绍常见的基于句法树和基于深度神经网络的语义角色标注算法。

4.5.1 基于句法树的语义角色标注方法

句法结构主要有成分结构和依存结构两大类。因此，依赖句法结构的语义角色标注算法可以进一步细分为：基于成分结构的语义角色标注（Span-Based SRL）和基于依存形式的语义角色标注（Dependency-Based SRL）。本节将针对上述两类方法分别进行介绍。

1. 基于成分句法树的语义角色标注方法

在基于成分结构的语义角色标注（Span-based SRL）中，模型基于句子的成分句法分析结果，对句中论元短语对应的跨度进行语义成分标注。例如，针对句子“美国波音公司正在制造民用飞机”，可以得到如图4.15所示成分句法分析结果，模型的目标是根据句子的句法成分标注，将名词短语“美国波音公司”，副词“正在”和名词短语“民用飞机”识别为谓词“制造”的施事者（A0）、受事

者（A1），以及表示时间关系的修饰语（AM-TMP）。

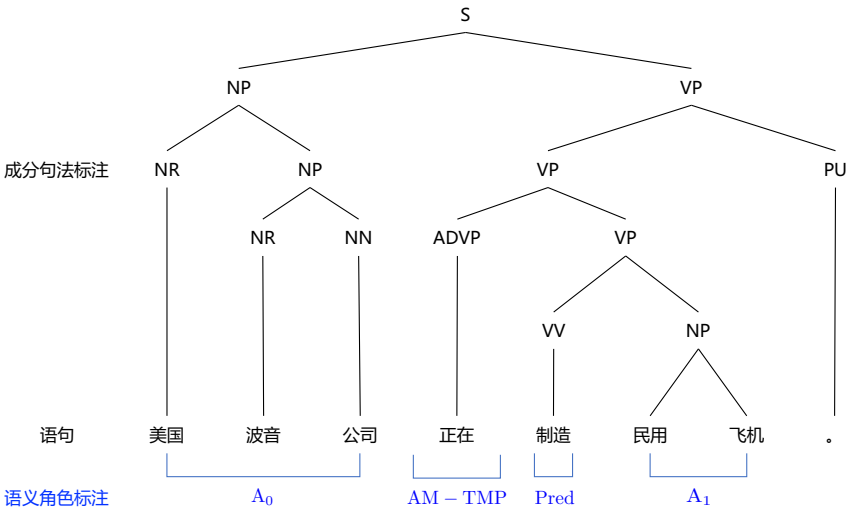

图 4.15 基于成分结构的语义角色标注

基于成分句法树的语义角色标注方法，通过对成分句法树进行剪枝，从句子中初步识别候选论元，供后续的论元识别、论元标注步骤使用。该方法的主要思想是考察句子中与谓词短语并列的成分，筛选符合条件的句子成分作为候选论元。具体而言，此方法从成分句法树的谓词结点开始，考察该结点的每个兄弟结点；如果兄弟结点和该结点在句法结构上不是并列关系，则将兄弟结点加入候选论元集合；如果兄弟结点是介词短语（PP），则将兄弟结点的全体子结点加入候选论元集合。依次地对谓词结点的父结点等每个祖先结点执行上述过程，直至到达根结点为止。以图4.16为例，自谓语（VV）“制造”开始，此方法逐次考察包含此谓语的谓语短语（VP），即“制造民用飞机”等成分。在此过程中，此方法将“民用飞机”、“正在”、“。”和“美国波音公司”加入候选论元集合，并过滤掉大量不可能是论元的成分。

在上述筛选过程后，训练分类模型从候选论元集合中识别真正的论元，并标注论元类型。在此过程中，通常需要为分类器构造有效的特征，常用特征可以分为以下类别[38]：

- 谓词及相关特征：谓词，谓词的语态，或论元和谓词出现的前后关系等。
- 论元的词特征：论元的中心词及其词性，以及头尾单词等。
- 基于成分句法标注的特征：论元的成分类型，树中论元到谓词的路径，成分的父亲、兄弟结点类型等。

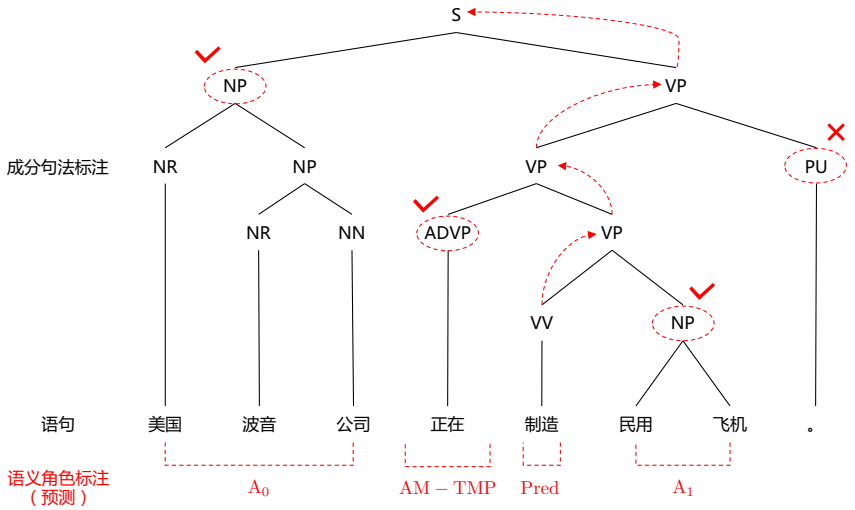

图 4.16　基于成分句法树的语义角色标注方法

2. 基于依存关系树的语义角色标注方法

在基于依存的语义角色标注（Dependency-based SRL）中，模型基于已知的句子依存分析树进行语义角色标注。如图4.17所示，给定句子“美国波音公司正在制造民用飞机”及其依存句法树，样本中标注谓词-论元关系，表示谓词与论元的中心词之间的语法关系。在依存句法树中，每个论元自身内部的语法结构由依存关系展示，而论元和谓词的语法关系体现为论元中心词和谓词之间的依存关系。对于论元“民用飞机”和谓词“制造”，两者之间的语法关系通过从“制造”指向“飞机”的边，体现为宾语（OBJ）关系。

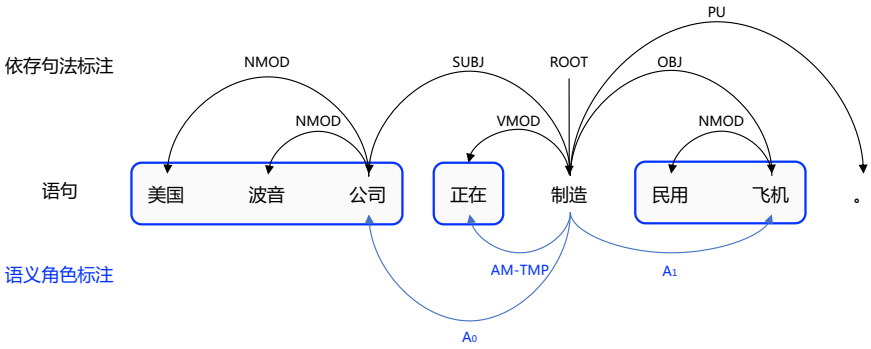

图 4.17　基于依存句法的语义角色标注

由于依存形式的语义角色标注把语义角色表示成谓词和论元中心词之间的语义关系，和依存句法标注完全对应，所以可以采用类似上节所述的剪枝方法，识别句子中潜在的论元。基于依存句法树的语义角色标注方法将上节所述的候选论元筛选过程迁移到依存句法树上。首先，从谓词结点开始，将当前结点的全体子结点加入候选论元集合；然后将当前结点的父结点作为当前结点，重复上述过程，逐次考察谓词结点的祖先结点；至当前结点为句子的根结点为止。如图4.18所示，谓语“制造”分别指向“公司”、“正在”、“飞机”和句尾的标点符号，对应的论元“美国波音公司”、“正在”和“民用飞机”将被识别为候选论元。

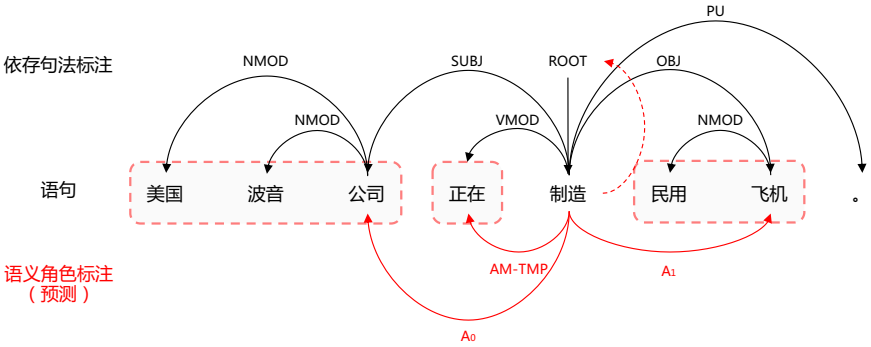

图 4.18　基于依存句法树的语义角色标注方法

针对后续的论元识别、论元标注阶段，基于依存句法树的语义角色标注方法将其建模为判断谓词和论元中心词之间语义关系的任务，并建立分类模型来解决。在此过程中常用的分类特征包括以下几类[38]：

- 谓词及相关特征：谓词，谓词的词根、词义、词性、语态，或论元和谓词出现的前后关系等。
- 论元的词特征：论元的中心词及其词性，以及头尾单词等。
- 基于成分句法标注的特征：树中论元中心词到谓词的路径，谓词与其父结点的依存关系，以及其父结点的相关信息；谓词与其子结点的依存关系；候选论元中心词的子结点、兄弟结点相关信息等。

4.5.2 基于深度神经网络的语义角色标注

在深度学习应用到自然语言处理领域后，由于模型能够自动学习到多层次的特征表示，不依赖句法标注、而直接对文本进行表示学习的方法也能达到较好的效果，并逐渐得到重视。针对语义角色标注而言，可以用 BIO 标注方案表示论元标签，从而可以直接利用通用的序列标注模型来解决；也可以以跨度标注句子中的论元短语位置，采用基于跨度预测的方法。由于跨度预测模型显式地建模了句子中短语级别的语义，模型可以更好地学习论元短语的长程邻接关系。因此，大量语义角色标注系统采用跨度预测的形式进行语义角色标注，并取得良好的效果。此外，由于句

法结构信息为语义角色标注任务提供了丰富的语言学信息，因此可有有效利用句法树结构的图神经网络也在该任务上取得了不错的效果。本节将分别介绍上述两类深度学习方法。

1. 基于跨度的语义角色标注方法

文献 [39] 中通过端到端模型构建了跨度预测模型，为语句中的每个词和跨度构造表示，实现同时识别句子中的谓词和论元并判断它们之间关系的效果。

该模型分为两个部分，词和跨度表示的构建以及谓词-论元的联合抽取。模型结构如图4.19表示，跨度预测模型包括一个编码器，用于为输入序列的每个标记构建表示。在文献 [39]，使用双向 LSTM 作为文本序列的编码器，编码器在每个位置的输入是单词的 GloVe 向量，以及通过 CharCNN 提取的字符级特征。针对每个单词 w_i，其嵌入表示为 $\boldsymbol{x}_i$=[WORDEMB(w_i); CHARCNN(w_i)]

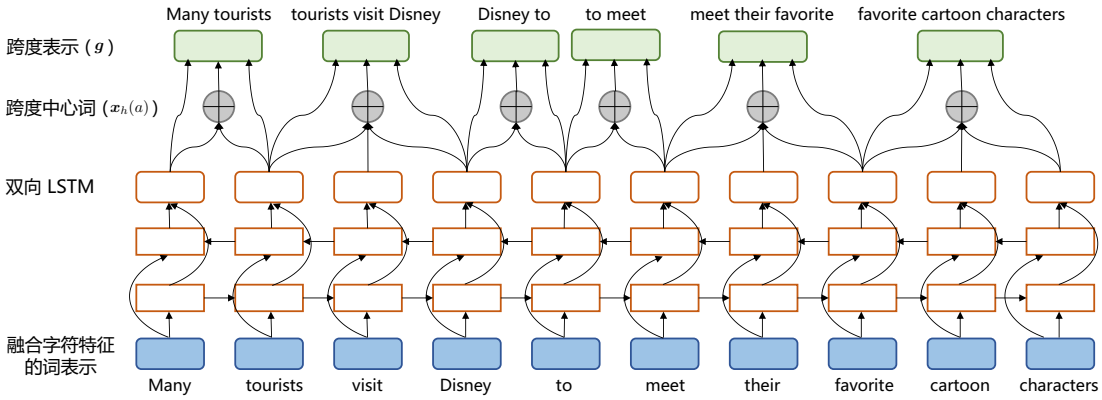

图 4.19　基于跨度预测的语义角色标注模型中跨度表示神经网络架构[39]

编码过程可以形式化的表示为：

$$\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n = \text{Encoder}(s = w_1, \ldots, w_n) \tag{4.65}$$

其中 $s = w_1, \ldots, w_n$ 是输入序列，$\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ 是对应位置的表示。

模型的词表示直接使用编码器在对应单词位置的输出，若潜在的谓词 $p = w_i$ 是句子中的第 i 个单词，则 $g(p) = \boldsymbol{v}_i$。跨度表示通过四部分构成，以 $a = (w_i, \ldots, w_j)$ 为例，分别是跨度内头尾标记的表示 $\boldsymbol{v}_i, \boldsymbol{v}_j$、中心词特征 $\boldsymbol{x}^h(a)$ 和跨度范围特征 $\phi(a)$，具体计算公式如下所示：

$$g(a) = [\boldsymbol{v}_i, \boldsymbol{v}_j, x^h(a), \phi(a)] \tag{4.66}$$

其中，中心词特征使用注意力机制建模了跨度内词语的重要程度，捕获句子中重要词在跨度内出

现的信息。对于跨度 $a = (w_i, \ldots, w_j)$, 中心词表示的计算如下所示:

$$e(a) = \text{softmax}(\boldsymbol{W}^e[\boldsymbol{v}_i; \ldots; \boldsymbol{v}_j]) \tag{4.67}$$

$$\boldsymbol{x}^h(a) = [\boldsymbol{x}_i; \ldots; \boldsymbol{x}_j]e(s)^T \tag{4.68}$$

其中, $\boldsymbol{W}^e$ 是可学习参数, 用于衡量单词的重要性; $e(s)$ 是句子 s 中每个词的注意力分数, $\boldsymbol{x}_i, \ldots, \boldsymbol{x}_j$ 是单词 $w_i, \ldots, w_j$ 经过编码器之前的原始表示, 由它们计算中心词特征。

跨度范围特征 $\phi(a)$ 仅和跨度的长度有关, 通常按照跨度的出现频率将跨度长度分为若干个范围, 为每个范围训练一个表示向量作为范围特征。在一般的语义角色标注任务中, 可以取 [1, 2, 3, 4, 5-7, 8-15, 16-31, 32-63, 64+] 作为跨度长度的区分范围。

在谓词-论元联合抽取部分, 通过计算谓词-论元匹配分数 $\Phi(l, a, p)$ 来识别谓词和论元, 以及论元相对谓词的语义角色, 其神经网络架构如图4.20所示。谓词-论元匹配分数主要通过组合谓词和论元的表示来计算:

$$\Phi_a(a) = \boldsymbol{W}^a \text{MLP}^a(g(a)) \tag{4.69}$$

$$\Phi_p(p) = \boldsymbol{W}^p \text{MLP}^p(g(p)) \tag{4.70}$$

$$\Phi_{rel}(l, a, p) = \boldsymbol{W}_l^r \text{MLP}^r([g(a); g(p)]) \tag{4.71}$$

$$\Phi(l, a, p) = \Phi_a(a) + \Phi_p(p) + \Phi_{rel}(l, a, p) \tag{4.72}$$

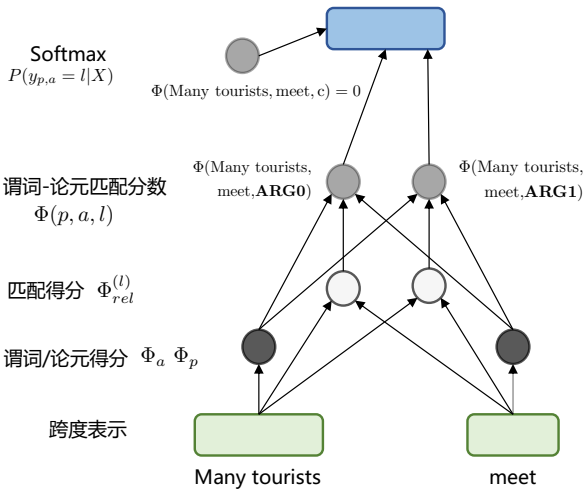

图 4.20 基于跨度预测的语义角色标注模型中谓词-论元联合抽取神经网络架构[39]

在句子长度为 n 时，潜在谓词和论元的个数分别为 $\mathcal{O}(n)$ 和 $\mathcal{O}(n^2)$ 个，如果要确定每个候选谓词和候选论元之间的语义角色关系，涉及的计算量为 $\mathcal{O}(n^3|\mathcal{L}|)$，其中 $\mathcal{L}$ 是标签集合。因此，使用束剪枝（Beam pruning）方法降低计算量，在计算谓词-论元匹配分数时，首先考察 $\Phi_a(a)$ 和 $\Phi_p(p)$，只保留分数较高的 $\lambda_a n$ 个论元和 $\lambda_p n$ 个谓词进行语义关系标签的计算，从而将计算量降至 $\mathcal{O}(n^2|\mathcal{L}|)$。

对于在剪枝中保留下来的谓词-论元对，计算匹配分数后，根据匹配分数的 Softmax 预测语义标签。在训练时，以负对数概率作为目标函数，最大化金标数据的出现概率。

$$P(y_{p,a} = l|X) = \text{softmax}(\Phi(l, a, p)) \tag{4.73}$$

$$P(Y|X) = \prod_{p,a} P(y_{p,a}|X) \tag{4.74}$$

$$\mathcal{L} = -\log P(Y^*|X) \tag{4.75}$$

2. 基于图卷积神经网络融合句法树的语义角色标注方法

语义角色标注与语法树有紧密的联系，如果能够在深度学习模型中融入语法树信息，能够带来效果提升。文献 [40] 提出了一种基于图网络的方法，可以有效融入依存句法树信息。该方法设计了一种结合门控机制的图网络，用于输入语句和句法树的编码。模型结构如图4.21所示。在传统的双向 LSTM 编码的基础上，该模型将 LSTM 编码层的输出接入一个图网络，通过在图网络中嵌入依存关系树，使语句编码获得语法信息。

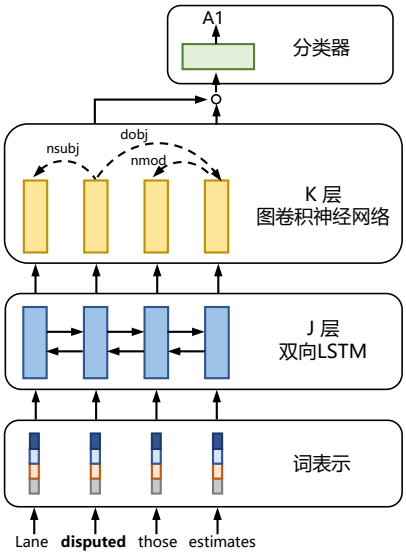

图 4.21　融合句法树的语义角色标注神经网络模型[40]

具体地，记 $\mathcal{G}=(\mathcal{V},\mathcal{E})$ 为句法树对应的图，其中结点集合 $\mathcal{V}$ 由句子中的每个词语组成，边集合 $\mathcal{E}$ 对应句法树中的依存弧。若边 $(u,v)\in\mathcal{E}$，记 $L(u,v)$ 为边 (u,v) 的类别标签，$D(u,v)$ 为边 (u,v) 的方向标签。方向标签的取值有三种，分别对应自指关系，以及依存句法树中的依存关系（双向）。若在依存关系树中存在由 u 指向 v 的弧，则 $(u,v)\in\mathcal{E}$ 且 $D(u,v)=0$，$(v,u)\in\mathcal{E}$ 且 $D(v,u)=1$。另外，对于 $\forall v\in\mathcal{V}$，$(v,v)\in\mathcal{E}$ 且 $D(v,v)=2$。类别标签包括方向信息和依存关系类型。易发现，$L(u,v)$ 和 $D(u,v)$ 分别对应边的细粒度和粗粒度类型标签。

在图网络的第 k 层，对于每个结点 v，通过 v 的邻接结点来更新 v 的表示向量：

$$\boldsymbol{h}_v^{(k+1)}=\mathrm{ReLU}\left(\sum_{u\in N(v)}\boldsymbol{g}_{v,u}^{(k)}(\boldsymbol{W}_{D(u,v)}^{(k)}\boldsymbol{h}_u^{(k)}+\boldsymbol{b}_{L(u,v)}^{(k)})\right) \tag{4.76}$$

其中，$\boldsymbol{h}_v^{(k+1)}$ 是结点 v 在第 $k+1$ 层的表示，$N(v)$ 是和 v 邻接的结点集合，$\boldsymbol{g}_{v,u}$ 是衡量 u 对 v 的重要性权重，$\boldsymbol{W},\boldsymbol{b}$ 是线性层的可学习权重，前者仅利用边的方向标签关系，从而大幅度减少了模型的参数规模。

门控机制通过为结点 v 的每个邻接结点 u 赋予权重，用来排除句法树中和谓词-论元关系无关的依存弧，也起到对依存句法分析结果进行去噪的效果。v 对 u 的重要性权重按照公式4.77进行计算：

$$\boldsymbol{g}_{u,v}^{(k)}=\sigma\left(\boldsymbol{W}_{D(u,v)}^{\prime(k)}\boldsymbol{h}_u^{(k)}+\boldsymbol{b}_{L(u,v)}^{\prime(k)}\right) \tag{4.77}$$

其中，$\boldsymbol{W}',\boldsymbol{b}'$ 是线性层的可学习权重。整体网络结构如图4.22所示。

在模型的整体训练过程中，输入序列首先经过双向 LSTM 编码，将编码结果作为图网络的结点初始表示，和句法树信息同时输入图网络。以图网络的输出作为句子中每个词的表示，该模型将数据标签以序列标注的形式进行编码，每个单词的类标签分布概率预测如下：

$$p(r|\boldsymbol{t}_i,\boldsymbol{t}_p,l)\propto exp(\boldsymbol{W}_{l,r}(\boldsymbol{t}_i\circ\boldsymbol{t}_p)) \tag{4.78}$$

$$\boldsymbol{W}_{l,r}=\mathrm{ReLU}(\boldsymbol{U}(\boldsymbol{q}_l\circ\boldsymbol{q}_r)) \tag{4.79}$$

其中 $\boldsymbol{t}_i,\boldsymbol{t}_p$ 是句子中第 i 个单词和谓词的输出表示，r 是语义角色标签，l 是谓词的词干，$\boldsymbol{U},\boldsymbol{q}$ 是可学习的权重。

4.5.3 语义角色标注评价方法

语义角色标注的评价指标包括精度（Precision，P）、召回率（Recall，R）和 F 值（F-Score）得分。在预测结果中，仅当论元范围和类型均预测正确时，才视为该论元预测正确，计为真正例（True-Positives, TP）。样本中标注，但未被模型预测正确的论元计为假反例（False-Negatives, FN）。

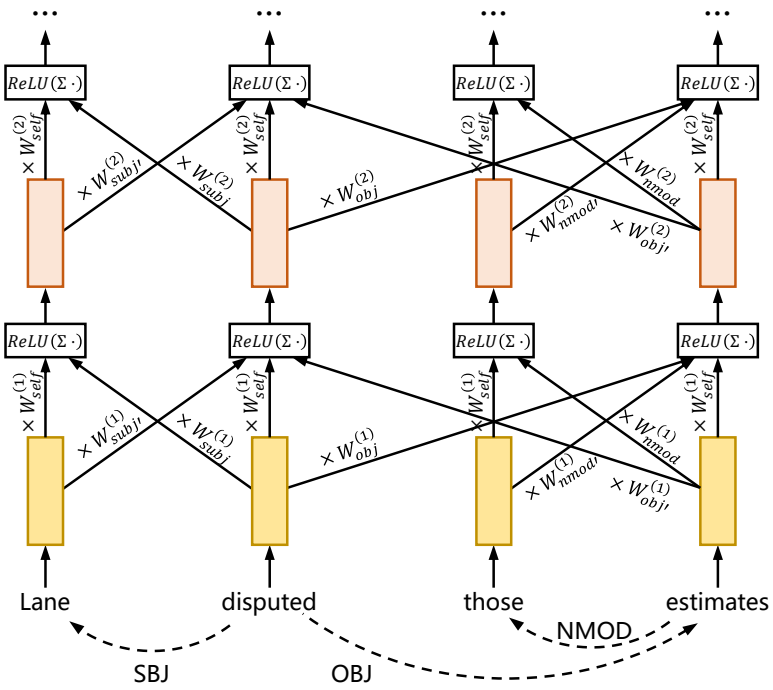

图 4.22 基于图网络的句法信息融合[40]

预测结果中出现，但无法与样本标注对应的论元预测结果计为假正例（False-Positives, FP）。语义角色标注的准确率为预测正确的论元数和预测结果中总的论元数之比值，即 $P = TP/(TP+FP)$. 召回率为预测正确的论元数和样本中标记的论元数之比值，即 $R = TP/(TP+FN)$. F1 得分为准确率、召回率的调和平均值，$F = 2PR/(P+R)$.

4.5.4 语义角色标注语料库

语义角色标注所采用的语义理论不同，比如题元理论中称为论元或题元，而格语法中则称为语义格，因此形成了多种不同的标注数据库，主要包含三个类型：FrameNet[5]、PropBank[41]、NomBank[42]。本节中将首先介绍上述三类语义角色标注语料库，在此基础上介绍基于成分句法和依存句法的语义角色标注评测集合。

1. 语义角色标注语料库

框架网络（FrameNet）根据框架语义学理论，对英国国家语料库进行标注。框架语义学理论提出了语义框架的概念，认为词语的语义反映为人类理解语言时在大脑中激活的认知结构，即语义框架。每个词语对应的语义框架由不同类型、数量的框架元素组成，用来体现和区分词语的语

义与功能。在 FrameNet 中，语义标注主要以框架的形式描述谓词的语义，并试图描述框架之间的关系。除主流的动词谓词外，FrameNet 也包括部分名词以及形容词谓词标注。FrameNet 目前包括超过 200,000 个人工标注的句子，在句子中标注目标谓词的语义框架、语义角色，以及每个语义角色在句法层面的短语类型和句法功能。此外，FrameNet 目前包含超过 1200 个语义框架，涵盖了动词及部分形容词和名词。FrameNet 的发布机构认为，基于 FrameNet 的语义角色标注能够在信息提取、机器翻译、事件识别、情感分析等领域起到帮助.

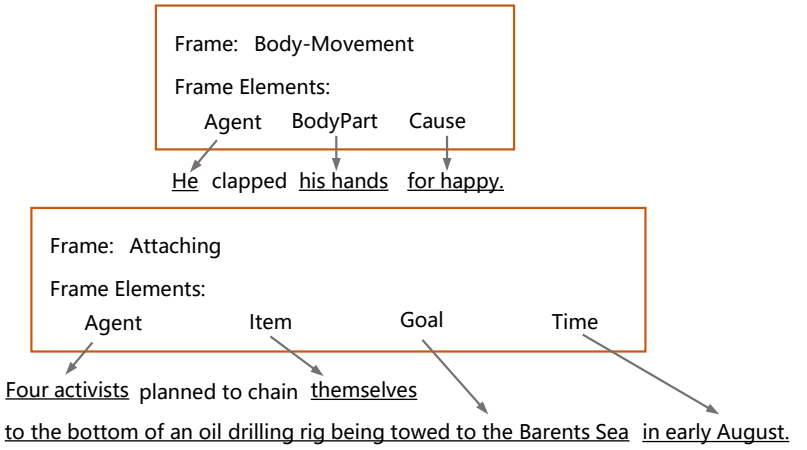

图 4.23 FrameNet 数据样例

命题库（Propositional Bank，PropBank）是论元角色语义知识库，针对动词性谓词，在语料中标注所在句的谓词-论元（predicate-argument）信息。PropBank 是基于英文宾州树库（Penn Tree Bank）标注，在英文宾州树库的句法结构标注基础上，标注谓词的论元及其语义角色。PropBank 对英文宾州树库中的 WSJ（华尔街日报标注）语料和一部分 BROWN（布朗语料库标注）语料进行了标注。

PropBank 定义了四大类的语义角色[43]：

- 核心语义角色：标记为 A0-A5 六种，如 A0 通常表示动作的施事者，A1 一般表示动作的影响，A2-A5 根据谓语动词有不同的语义含义。
- 修饰作用的附加语义角色：其角色标签以 AM 开头，常见的 15 种有：ADV（附加的，默认标记）、BNE（受益人）、CND（条件）、DIR（方向）、DGR（程度）、EXT（扩展）、FRQ（频率）、LOC（地点）、MNR（方式）、PRP（目的或原因）、TMP（时间）、TPC（主题）、CRD（并列参数）、PRD（谓语动词）、PSR（持有者）和 PSE（被持有）。
- 参考语义角色：其角色标签以 R 开头，表示出现在句子中的其他论元。

• 动词：其角色标签为 V，表示句子中的谓语动词。

名词命题库（NomBank）关注并侧重名词性谓词的语义角色标注，对 PropBank 的涵盖范围进行了补充。在标注句子中的谓词时，PropBank 是基于动词词典进行标注的，只考虑动词性谓词，而未涉及语料中的名词性谓词。针对这项不足，NomBank 对语料中的名词性谓词进行语义角色标注。其主要语料资源同样来自《华尔街日报》，研究人员对其中名词性的谓词对应的论元角色进行人工标注。在 NomBank 的标注过程中，标注者尽可能地使角色定义在词性之间保持一致。例如，NomBank 在名词“decision”的标注中依然使用 PropBank 对动词“decide”的标注框架文件。图4.24给出了 NomBank 的数据样本示例。在名词短语“John’s replacement Ben”和“Ben’s replacement of John”中，名词“replacement”是谓词，Ben 是 Arg0，表示替代者；John 是 Arg1，表示被替代者。

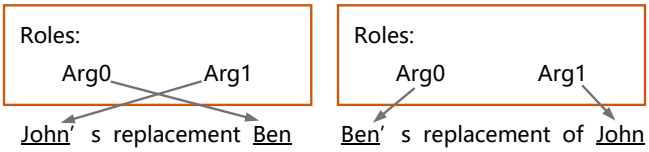

图 4.24　NomBank 数据样例

NomBank 对名词性谓词的语义解析是对该领域语言资源的一项重要补充。根据标注者的统计，对于 NomBank 所标注的伴随论元结构的名词，其中约有一半是名词化的，或者具有类似名词化的性质。例如，”aggression” 和”agenda” 具有类似动词”destroy” 和”schedule” 的论元结构。NomBank 的标注工作揭示了名词论元结构的一系列语言现象，包括支持动词结构、跨系论元、和括号内的 PP 结构等。由于名词性谓词的论元可以出现在以该名词为首的 NP 之外，这些语言现象都具有较高的研究价值。

2. 基于成分结构的语义角色标注评测

在基于成分结构的语义角色标注中，常用的评测基准是 CoNLL05 和 CoNLL12 数据集，其标注格式如4.5.1节的图4.15所示。CoNLL05 数据集主要由英文宾州树库的 WSJ 部分组成，语料来自《华尔街日报》的新闻报道，数据标注包括宾州树库的句法树标注，以及从 PropBank 中提取的谓词-参数结构信息。在该基准中，遵循句法解析中使用的标准分区，WSJ 语料的第 02-21 节用作训练集，第 24 节用作验证集，第 23 节用作测试集。此外，CoNLL05 的测试集还包括一个领域外数据集，语料来自 Brown 语料库的 ck01-03 三个部分，类型为小说文本，用于测试模型对领域外样本的泛化能力。

在 OntoNotes 语料库提出后，由于其规模大并且涉及多个领域，还具有多种语言等特点，CoNLL 基于该语料库提出了语义角色标注的 CoNLL12 基准。CoNLL12 的标注形式和标注方式和 CoNLL05 接近，主要的改进体现在更大的规模和更广泛的语料来源上。另外，CoNLL12 包含多种语言的评

测基准。表4.11展示了 CoNLL05 及 CoNLL12 评测基准的统计数据，包括 CoNLL12 的中文评测基准[44]。

表 4.11 CoNLL05、CoNLL12 评测基准的统计数据

数据集	CoNLL 2005				CoNLL 2012（英文）			CoNLL 2012（中文）		
数据划分	训练集	验证集	测试集	Brown	训练集	验证集	测试集	训练集	验证集	测试集
句子个数	39.8k	1.3k	2.4k	0.4k	75.2k	9.6k	9.5k	36.5k	6.1k	4.5k
谓词个数	90.8k	3.2k	5.3k	0.8k	188.9k	23.9k	24.5k	117.1k	16.6k	15.0k
论元个数	333.7k	11.7k	19.6k	3.0k	622.5k	78.1k	80.2k	365.3k	51.0k	46.7k

3. 基于依存结构的语义角色标注评测

CoNLL09 是目前常用于依存形式语义角色标注的评测基准。CoNLL09 数据集包含来自不同语系的加泰罗尼亚语、汉语、捷克语、英语、德语、日语和西班牙语等 7 种语言，用于评估系统在给定谓词的情况下，进行谓词消歧、论元识别和论元分类的能力。CoNLL09 的英语部分主要基于 PropBank 和 NomBank 的标注，包含动词性和名词性谓语，训练集和验证集分别包含 39.3k 和 2.4k 个句子。汉语部分主要基于汉语树库和汉语命题库的标注，训练集和验证集分别包含 22.3k 和 2.6k 个句子，具体的统计数据如表4.12所示[45, 46]。由于语言内容的丰富性，CoNLL09 经常用于评估语义角色标注模型的多语言扩展能力。另外，由于 CoNLL09 在捷克语、英语和德语上准备了领域外测试集，其也经常用于测试语义角色标注模型的泛化性能。

表 4.12 CoNLL09 评测基准的统计数据

数据集	训练集句数	训练集词数	平均句长	谓词比例	验证集句数	验证集词数
CoNLL 2009（英文）	39.3k	958.2k	24.4	18.7%	2.4k	57.7k
CoNLL 2009（中文）	22.3k	609.1k	27.3	16.9%	2.6k	73.2k

4.6 延伸阅读

4.7 习题

(1) 词汇语义关系在自然语言处理的下游任务中有哪些应用？试结合 WordNet 举例说明。
(2) 如何在深度学习模型中融合句子的谓词逻辑表示式？
(3) 语言的分布式表示和语言模型之间有什么区别和联系？
(4) 除了词义相似性和词的类比性之外，词向量还能体现哪些单词的语义性质？试结合 t-SNE 可视化分析举例说明。
(5) 如何在对话系统中应用词义消歧方法，提升综合性能？

(6) 少样本、资源受限场景下如何进行语义角色标注?

参考文献

[1] 何三本, 王玲玲. 现代语义学[M]. 台北: 三民书局, 1995.

[2] Wierzbicka A. Semantic primitives[J]. 1972.

[3] Dong Z, Dong Q. Hownet-a hybrid language and knowledge resource[C]//International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003. IEEE, 2003: 820-824.

[4] Fillmore C J, Atkins B T. Toward a frame-based lexicon: The semantics of risk and its neighbors[J]. Frames, fields and contrasts: New essays in semantic and lexical organization, 1992, 75:102.

[5] Baker C F, Fillmore C J, Lowe J B. The berkeley framenet project[C]//COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics. 1998.

[6] 李福印. 语义学概论[M]. 北京大学出版社, 2006.

[7] Miller G A. Wordnet: a lexical database for english[J]. Communications of the ACM, 1995, 38(11): 39-41.

[8] 梅德明. 语言学与应用语言学百科全书[M]. 北京大学出版社;, 2017.

[9] Sundheim B M. Overview of the third message understanding evaluation and conference[C]// Proceedings of the 3rd conference on Message understanding. 1991: 3-16.

[10] Hendrix G G. Expanding the utility of semantic networks through partitioning[C]//Proceedings of the 4th international joint conference on Artificial intelligence-Volume 1. 1975: 115-121.

[11] Salton G, Wong A, Yang C S. A vector space model for automatic indexing[J]. Communications of the ACM, 1975, 18(11):613-620.

[12] Bengio Y, Ducharme R, Vincent P. A neural probabilistic language model[J]. Advances in neural information processing systems, 2000, 13.

[13] Almeida F, Xexéo G. Word embeddings: A survey[J]. arXiv preprint arXiv:1901.09069, 2019.

[14] Sahlgren M. The distributional hypothesis[J]. Italian Journal of Disability Studies, 2008, 20:33-53.

[15] Deerwester S, Dumais S T, Furnas G W, et al. Indexing by latent semantic analysis[J]. Journal of the American society for information science, 1990, 41(6):391-407.

[16] Levy O, Goldberg Y, Dagan I. Improving distributional similarity with lessons learned from word embeddings[J]. Transactions of the association for computational linguistics, 2015, 3:211-225.

[17] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [C]//ICLR. 2013.

[18] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.

[19] Sennrich R, Haddow B, Birch A. Neural machine translation of rare words with subword units[C]// 54th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (ACL), 2016: 1715-1725.

[20] Agirre E, Alfonseca E, Hall K, et al. A study on similarity and relatedness using distributional and wordnet-based approaches[J]. 2009.

[21] Hill F, Reichart R, Korhonen A. SimLex-999: Evaluating semantic models with (genuine) similarity estimation[J/OL]. Computational Linguistics, 2015, 41(4):665-695. https://aclanthology.org/J15-4004. DOI: 10.1162/COLI$_{a0}$0237.

[22] Kiros R, Zhu Y, Salakhutdinov R R, et al. Skip-thought vectors[C/OL]//Cortes C, Lawrence N, Lee D, et al. Advances in Neural Information Processing Systems: volume 28. Curran Associates, Inc., 2015. https://proceedings.neurips.cc/paper/2015/file/f442d33fa06832082290ad8544a8da27-Paper.pdf.

[23] Pagliardini M, Gupta P, Jaggi M. Unsupervised learning of sentence embeddings using compositional n-gram features[C/OL]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics, 2018: 528-540. https://aclanthology.org/N18-1049. DOI: 10.18653/v1/N18-1049.

[24] Bojanowski P, Grave E, Joulin A, et al. Enriching word vectors with subword information[J]. Transactions of the Association for Computational Linguistics, 2016.

[25] Joulin A, Grave E, Bojanowski P, et al. Bag of tricks for efficient text classification[C/OL]// Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. Valencia, Spain: Association for Computational Linguistics, 2017: 427-431. https://aclanthology.org/E17-2068.

[26] Gale W A, Church K W, Yarowsky D. A method for disambiguating word senses in a large corpus [J]. Computers and the Humanities, 1992, 26(5):415-439.

[27] Melamud O, Goldberger J, Dagan I. context2vec: Learning generic context embedding with bidirectional lstm[J]. conference on computational natural language learning, 2016.

[28] Peters M, Neumann M, Iyyer M, et al. Deep contextualized word representations[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers): volume 1. 2018: 2227-2237.

[29] Loureiro D, Jorge A. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation[C/OL]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 5682-5691. https://aclanthology.org/P19-1569. DOI: 10.18653/v1/P19-1569.

[30] Blevins T, Zettlemoyer L. Moving down the long tail of word sense disambiguation with gloss-informed biencoders[J]. arXiv preprint arXiv:2005.02590, 2020.

[31] Huang L, Sun C, Qiu X, et al. GlossBERT: BERT for word sense disambiguation with gloss knowledge[C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 3509-3514. https://aclanthology.org/D19-1355. DOI: 10.18653/v1/D19-1355.

[32] Levine Y, Lenz B, Dagan O, et al. SenseBERT: Driving some sense into BERT[C/OL]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, 2020: 4656-4667. https://aclanthology.org/2020.acl-main.423. DOI: 10.18653/v1/2020.acl-main.423.

[33] Loureiro D, Jorge A. Language modelling makes sense: Propagating representations through wordnet for full-coverage word sense disambiguation[J]. arXiv preprint arXiv:1906.10007, 2019.

[34] Miller G A, Chodorow M, Landes S, et al. Using a semantic concordance for sense identification[C]// Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994. 1994.

[35] Petrolito T, Bond F. A survey of WordNet annotated corpora[C/OL]//Proceedings of the Seventh Global Wordnet Conference. Tartu, Estonia: University of Tartu Press, 2014: 236-245. https://aclanthology.org/W14-0132.

[36] Taghipour K, Ng H T. One million sense-tagged instances for word sense disambiguation and induction[C]//Proceedings of the nineteenth conference on computational natural language learning. 2015: 338-344.

[37] Raganato A, Camacho-Collados J, Navigli R. Word sense disambiguation: A unified evaluation framework and empirical comparison[C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. 2017: 99-110.

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

[39] He L, Lee K, Levy O, et al. Jointly predicting predicates and arguments in neural semantic role labeling[C/OL]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Melbourne, Australia: Association for Computational Linguistics, 2018: 364-369. https://aclanthology.org/P18-2058. DOI: 10.18653/v1/P18-2058.

[40] Marcheggiani D, Titov I. Encoding sentences with graph convolutional networks for semantic role labeling[C/OL]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, 2017: 1506-1515. https://aclanthology.org/D17-1159. DOI: 10.18653/v1/D17-1159.

[41] Palmer M, Gildea D, Kingsbury P. The proposition bank: An annotated corpus of semantic roles[J]. Computational linguistics, 2005, 31(1):71-106.

[42] Meyers A, Reeves R, Macleod C, et al. The nombank project: An interim report[C]//Proceedings of the workshop frontiers in corpus annotation at hlt-naacl 2004. 2004: 24-31.

[43] Carreras X, Màrquez L. Introduction to the CoNLL-2004 shared task: Semantic role labeling[C/OL]//Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004. Boston, Massachusetts, USA: Association for Computational Linguistics, 2004: 89-97. https://aclanthology.org/W04-2412.

[44] Zhang Z, Strubell E, Hovy E. Comparing span extraction methods for semantic role labeling[C/OL]//Proceedings of the 5th Workshop on Structured Prediction for NLP (SPNLP 2021). Online: Association for Computational Linguistics, 2021: 67-77. https://aclanthology.org/2021.spnlp-1.8. DOI: 10.18653/v1/2021.spnlp-1.8.

[45] Hajivc J, Ciaramita M, Johansson R, et al. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages[C/OL]//Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task. Boulder, Colorado: Association for Computational Linguistics, 2009: 1-18. https://aclanthology.org/W09-1201.

[46] Li Z, Zhao H, Zhou J, et al. Dependency and span, cross-style semantic role labeling on propbank and nombank[J/OL]. ACM Trans. Asian Low-Resour. Lang. Inf. Process., 2022. https://doi.org/10.1145/3526214.

索引