自然语言处理导论

张奇 桂韬 黄萱菁

2022 年 12 月 23 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

7. 信息抽取

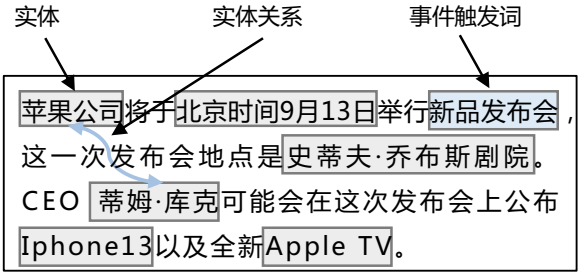
随着互联网的迅猛发展，大量的信息以电子文档的形式出现，人们面临的不再是信息匮乏，而是严重的信息过载。为了应对信息爆炸带来的挑战，迫切需要一些自动化的工具将大量无结构的文本内容及时准确地进行抽取、过滤、归类组织，帮助人们在海量内容中迅速找到真正需要的信息。信息抽取就是在这样的需求下应用而生。**信息抽取**（Information Extraction，IE）任务的目标就是从非结构化的文本内容中提取特定的信息。信息抽取并不试图对全文进行理解，仅针对任务需求和目标从篇章中抽取特定信息。信息抽取的应用广泛，在阅读理解、机器翻译、知识图谱等任务中都发挥着非常基础和重要的作用。

本章首先介绍信息抽取的基本概念，并详细介绍信息抽取的三个主要任务：命名实体识别、关系抽取和事件抽取，在此基础上介绍不同场景下信息抽取的难点及主要算法。

7.1 信息抽取概述

海量的文本内容提供了人们丰富的信息获取的可能，但是面对如此巨量的信息，人们也难以快速从这些内容发现所需的信息。迫切需要自然语言处理算法能够自动化的从这些无结构的文本中发现特定信息。但是通过第 4 章语义分析的介绍我们可以知道，通用的句子和篇章的语义表示和理解目前还远达不到实用的阶段。信息抽取目标不是构建通用的句子或者篇章理解，而是针对特定的需求，从自然语言构成的非结构化文本中抽取指定类型的实体、关系、事件等信息，进而形成结构化数据。信息抽取是自然语言处理任务中重要的研究方向和底层任务。

如图7.1所示，信息抽取系统可以从一段非结构化的新闻文本抽取出公司：“苹果公司”、时间：“北京时间 9 月 13 日”、地点：“史蒂夫·乔布斯剧院” 等实体信息，以及 “蒂姆·库克” 与 “苹果公司” 是 “CEO-Of” 关系的信息，并且还可以获得整段文本描述的是 “发布会” 事件。抽取后的数据和事实可以直接向用户显示，也可作为原文检索的索引，或存储到数据库、电子表格中，以便于以后的进一步分析。例如，从新闻报道中抽取出恐怖事件的详细情况：时间、地点、作案者、受害者、袭击目标、使用的武器等；从经济新闻中抽取出公司发布新产品的情况：公司名、产品名、发布时间、产品性能等；从病人的医疗记录中抽取出症状、诊断记录、检验结果、处方等。使用信息抽取方法所获的信息构成结构化描述，可以直接存入数据库中，供用户查询进一步分析利用。



事件类型	发布会
时间	北京时间9月13日
公司	苹果公司
地点	史蒂夫·乔布斯剧院
人员	蒂姆·库克
产品	Iphone13、Apple TV
实体关系	苹果公司，蒂姆·库克，CEO

图 7.1　非结构化文本信息抽取样例

信息抽取技术属于知识技术中知识发现的范畴，信息抽取的宗旨在于抽取指定的信息，它突破了信息检索中必须由人来阅读、理解、抽取信息的局限性，实现了信息的自动查找、理解和抽取。一个好的信息抽取模型可以极大地促进下游自然语言处理任务性能的提高。实体、关系、事件作为文本中重要的语义知识，可以为信息检索、知识图谱、问答系统等提供基础支撑。例如，实体及关系可以改善系统检索文档的相关度，并提高检索系统的召回率和准确率；实体、关系及事件等是知识图谱的基本元素；实体与关系可以支持问答系统对文本中的关键信息做出更准确的分析，给出更精确、更简洁的短语级的答案。

一般来说，信息抽取系统的处理对象是自然语言文本，尤其是非结构化文本。但从广义上讲，除了电子文本以外，信息抽取系统的处理对象还可以是语音、图像、视频等其他媒体类型的数据。本章只讨论狭义上的信息抽取，即针对自然语言文本的信息抽取。

信息抽取研究开始于 20 世纪 60 年代中期，以纽约大学的 Linguistic String 和耶鲁大学的 FRUMP 这两个长期项目为代表。直到 20 世纪 80 年代末期，得益于消息理解系列会议（Message Understanding Conference，MUC）的举办，信息抽取的研究与应用逐步进入繁荣期。从 1987 年到 1998 年，MUC 会议共举行了 7 届，MUC 为信息抽取制定了具体的任务和严密的评测体系，该会议提出了一套完整的基于模板填充机制的信息抽取方案，核心内容包括命名实体识别、共指消解、关系抽取、事件抽取等具体内容。该会议吸引了世界各地的研究者参与其中，从理论和技术上促进了信息抽取的研究成果不断涌现。MUC 为信息抽取在 NLP 领域中成为一个独立分支做出了重大贡献。

命名实体是在 1995 年的 MUC-6 信息理解会议中首次提出，其主要的研讨内容为如何从论文、新闻报纸等非结构化文本中抽取公司、国防等相关信息，这些信息包括了现在数据集中常用的人名、地名、机构名等标签。MUC-6 会议评测任务就是自动识别文本中的预定义的实体并进行分类，但当时的研究方法基本上是基于模板规则，比如词汇规则（包括词形，词性），短语规则等。因为测试语料主题单一且数量较小（30 篇），因此大部分的识别方法都取得了较好的成绩，最高的 F1

值达到 96.42%。除了实体识别任务，MUC 会议还引入三个新的评测任务：共指（关系确定、模板元素填充等。随后的 MUC-7 会议拓展了 MUC-6 的标注规范，并且增加了训练语料的数量。值得注意的是，MUC-7 中出现了隐马尔可夫，最大熵等基于统计机器学习的方法。MUC-6 和 MUC-7 中将有待识别的实体指称为 “实体的唯一标识符（Unique identifiers of entities）”，目标实体类型分为三类：命名实体、时间、数值，其中命名实体又可细分为：人名、地名、机构名，但是缺少对命名实体的深入讨论和定义。

继 MUC 会议之后，1999 年至 2008 年美国国家标准技术研究所（NIST）组织的自动内容抽取（Automatic Content Extraction，ACE）评测会议成为另一个致力于信息抽取研究的重要国际会议。与 MUC 相比，ACE 评测不针对某个具体的领域或场景，它采用基于漏报（标准答案中有而系统输出中没有）和误报（标准答案中没有而系统输出中有）的一套评价体系，还对系统跨文档处理（Cross-document Procesing）能力进行评测。这一新的评测会议把信息抽取技术研究引向新的高度。

除了 MUC 和 ACE 外,还有多语种实体评价任务会议（Multilingual Entity Task Evaluation,MET)、文本理解会议（Document Understanding Conference，DUC）等与信息抽取相关的国际学术会议，它们为信息抽取在不同领域、不同语言中的应用起到了很大的推动作用。

中文的信息抽取研究起步相对英文较晚，由于中文与西方字母型文字的巨大差异，以及中文缺少表示词语边界的分割符号等特殊性，中文信息抽取效果会受到自动分词结果的影响，导致中文信息抽取研究进展较慢。早期工作主要集中在中文命名实体识别方面，在 MUC-7、MET 等会议的支持下，取得了一定的进步。2006 年 SIGHAN（Special Interest Group of the Association for Computational Linguistics）将汉语命名实体识别加入 Bakeoff 评测比赛。Bakeoff-2006 遵循了 CoNLL-2002 的标签定义框架，总共提出并标注了四种命名实体标签，包括人名、地名、机构名和地缘政治实体，并提供了三个中文语料库：MSRA，LDC 和 CITYU。Bakeoff-2007 删除了 LDC 语料库，并将命名实体类型设置为人名，地点和机构名。在 Bakeoff-2006 和 Bakeoff-2007 中，研究者多使用了统计机器学习方法，并且最优秀的 NER 方法几乎都使用了条件随机场、最大熵等统计机器学习模型。当前中文信息抽取研究在继续优化命名实体识别效果的基础上，已经向着共指消解、关系抽取、事件抽取等更高阶段发展。虽然当前信息抽取通常还只是面向特定领域开展，能够真正实现大规模应用的通用信息抽取系统仍然未出现，但是应当看到，近年来信息抽取领域呈现出更为活跃的态势，从理论到应用都有一些新进展。

信息抽取包含命名实体识别（Named Entity Recognition，NER）、关系抽取（Relation Extraction，RE）、事件抽取（Event Extraction）、时间表达式识别（Temporal Expression）、实体归一化（Entity Normalization）、模板填充（Template Filling）、话题检测于跟踪（Topic Detection and Tracking，TDT）等任务。由于这些任务是大多数自然语言处理系统所依赖的底层工具，因此自 21 世纪以来，不论在学术界还是工业界都都对信息抽取任务的研究，给与了越来越多的关注。本章主要对命名实体识别、关系抽取以及事件抽取任务和常见算法进行介绍。

7.2 命名实体识别

命名实体（Named Entity）是指具有特定意义的实体，主要包括人名、地名、机构名、专有名词等。常见的命名实体和样例如表7.1所示。**命名实体识别**（Named Entity Recognition，NER）目标就是从文本中抽取出这些具有特定意义的实体词。命名实体识别一般包含两个步骤，分别是实体边界判断和实体类别判断。其中实体边界判断是为了确定实体字符串在非结构化文本中的开始位置和结束位置，而实体类别的判断则是为了判断该字符串对应的实体类型。

例如：复旦大学始创于 1905 年，原名复旦公学，1917 年定名为复旦大学，位于中国上海，是中国人自主创办的第一所高等院校。

其中“复旦大学”和“复旦公学”是机构名，“1905”和“1917”是时间，“上海”是地名。

表 7.1 常见命名实体以及样例

实体名	标签	举例
人名	PER	[张钹]$_{PER}$ 院士：抓住机会、掌握主动发展第三代人工智能
地名	LOC	2021 世界人工智能大会将于 7 月 8 日至 10 日在 [上海]$_{LOC}$ 召开
机构名	ORG	[复旦大学]$_{ORG}$ 校名取自《尚书大传》之“日月光华，旦复旦兮”
疾病名	DIS	[高血压]$_{DIS}$ 已成为影响全球死亡率的第二大危险因素
药品名	DRU	[奥司他韦]$_{DRU}$ 是治疗流感的首选药物

命名实体识别算法的主要难度在于处理歧义和未登录词问题。歧义问题是指同一个名称可以指代不同类型的实体。

例如：我们明天在 [复旦大学] 见。

[复旦大学] 共有邯郸、枫林、张江、江湾四个校区。

这里“复旦大学”在不同的上下文中分别作为地名和机构名。在英文中这种现象更加常见，比如：“Harvard”既可以是人名，也可以是机构名，还可以是地名，需要根据上下文对实体的类别进行判断。未登录词问题与中文分词中定义一致，也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实体。命名实体在语言中通常表现出表达随意、用法复杂、形式多变等特点，未登录词问题相较于中文分词更加严重。

命名实体从表现形式还可以进一步分为两种类型：**非嵌套命名实体**（Non-nested Named Entities）和**嵌套命名实体**（Nested Named Entities）。非嵌套命名实体就是普通的命名实体，每个单词只对应一个标签；嵌套命名实体是指实体中存在嵌套的情况，每个的单词可能对应若干个标签。

例如：“复旦大学上海医学院”整体上是机构名

“上海”是地名，“复旦大学”是机构名

具有嵌套结构的命名实体，传统的基于序列标注的命名实体模型是难以直接有效地处理的。在本节中，我们将对这两种实体类型的识别方法分别进行介绍。

7.2.1 非嵌套命名实体识别

非嵌套命名实体识别通常可以转换为序列标注问题。对序列的每一个元素（Token）标注一个标签。一般来说，序列通常是一个句子，而元素指的是句子中的词语或者字。标注的标签一般同时能表示实体的边界和类别信息。典型的标注格式是 *BIO* 标签体系，即将每个元素标注为“B-X”、“I-X”或者“O”。其中，“B-X”表示此元素所在的片段属于 X 类型并且此元素在此片段的开头，“I-X”表示此元素所在的片段属于 X 类型并且此元素在此片段的中间位置，“O”表示不属于任何类型。假设需要识别的命名实体包含人名、地名和机构名，采用 *BIO* 标签体系，对应的标签集合为：{O, B-PER, I-PER, B-LOC, I-LOC, B-ORG, I-ORG}。图7.2给出了利用该标签体系对于句子中每个词语的分类标签。对于句子“Democrat Biden replaces Trump as President of the United States”，包含四个命名实体：“Democrat”是组织名，“Biden”和“Trump”都是人名，“United States”是地名。

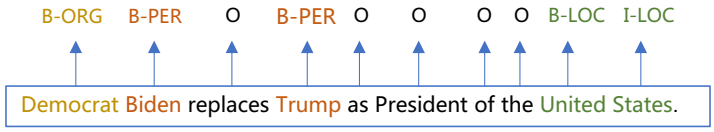


图 7.2 非嵌套命名实体识别示例

在实际应用中，也有一些系统采用更复杂的 *BIOES* 标签体系，在 *BIO* 标签的基础上增加了单字符实体 *S* 和字符实体的结束标识 *E*。同样的，在边界标签基础上需要与实体类别标签融合，对应的标签集合为 {O, B-PER, I-PER, E-PER, S-PER, B-LOC, I-LOC, E-LOC, S-LOC, B-ORG, I-ORG, E-ORG, S-ORG}。具体来说，对于一个长度为 n 的句子 $S = \{w_1, w_2, \ldots, w_n\}$，其中 w_i 表示句子中的第 i 个元素。序列标注即给每个元素赋予一个标签，来表示这个元素所在的实体类别和边界信息。一组相邻且类别相同的元素构成的子序列 $span(start, end, type)$ 就构成了抽取出来的命名实体。通过这种转换，命名实体识别问题就转换为了序列标注问题。

本节将介绍基于半马尔可夫条件随机场和基于 Transformer 方法的命名实体算法，以及融合字典信息的中文命名实体识别算法。

1. 基于半马尔可夫条件随机场的命名实体识别

非嵌套的命名实体识别任务通过上述 BIO 标签，可以转换为序列标注任务。因此，可以使用包括隐马尔可夫（HMM），条件随机场（CRFs）等在序列标注模型进行建模，相关工作可以参考本书第 2 章相关内容。在自然语言处理算法中通常使用的线性链条件随机场针对随机变量序列 X 的条件下，随便变量序列 Y 的条件概率分布 $P(Y|X)$ 满足马尔可夫性，即输出标签序列 y_i 仅与其周边的标签 y_{i-1} 和 y_{i+1} 相关。在输入序列 X 为句子中每个字的情况下，使用线性链条件随机场仅能表示局部依赖的特征。但是，在命名实体识别任务中往往依赖更多的非局部特征。

2004 年 Sarawagi 和 Cohen 提出了半马尔可夫条件随机场（Semi-Markov Conditional Random Fields，Semi-CRFs）[1]，从要求每个字的所对应分类标签满足马尔可夫性，放松到仅需由邻接词组成的片段（Segments）间进行满足马尔可夫性即可。将这种模型用于命名体识别任务时，可以更有效、更自由的利用各种有利于识别出命名体片段边界的特征，如实体的长度、与实体相似的已知实体名称等。

在建立一个条件随机场时，首先要定义一个特征函数集，该函数集内的每个特征函数都以文本作为输入，提取的特征作为输出，假设该函数集为:

$$\Phi(x_1 \dots x_n; y_1 \dots y_n) \tag{7.1}$$

其中 $\boldsymbol{x} = \{x_1 \dots x_n\}$ 为 n 个字组成的输入文本序列，$\boldsymbol{y} = \{y_1 \dots y_n\}$ 为对应的实体标签序列。条件随机场使用对数线性模型来计算给定观测序列下状态序列的条件概率：

$$P(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{w}) = \frac{\exp(\boldsymbol{w} \cdot \Phi(\boldsymbol{x}, \boldsymbol{y}))}{\sum_{\boldsymbol{y}'} \exp(\boldsymbol{w} \cdot \Phi(\boldsymbol{x}, \boldsymbol{y}'))} \tag{7.2}$$

其中 $\boldsymbol{y}'$ 是所有可能的状态序列，$\boldsymbol{w}$ 是 CRF 模型的参数，可以把它看成是每个特征函数的权重，CRF 模型的训练可以看作是对参数 $\boldsymbol{w}$ 的估计。

当扩展至半马尔可夫条件随机场时，使用片段集合 $s =< s1, \dots, s_p >$ 来表示输入文本 $\boldsymbol{x}$，其中一个片段 $s_j =< t_j, u_j, y_j >$ 由起始位置 t_j、结束位置 u_j 和标签 $y_j \in \boldsymbol{y}$ 组成。这里所有片段的长度都为正数，并且完全地覆盖序列 X 且没有重叠，也就是说 t_j 和 u_j 总是满足以下的约束：

$$\begin{aligned} t_1 = 1, u_p = |\boldsymbol{x}| \\ 1 \leqslant t_j \leqslant u_j \leqslant |\boldsymbol{x}| \\ t_{j+1} = u_j + 1 \end{aligned} \tag{7.3}$$

例如：对于输入文本序列“I went skiing with Fernando Pereira in British Columbia”，一种可能得片段表示为 $\boldsymbol{s} =< (1,1,O), (2,2,O), (3,3,O), (4,4,O), (5,6,I), (7,7,O), (8,9,I) >$，对应标签序列 $\boldsymbol{y} =< O, O, O, O, I, I, O, I, I >$。根据公式7.2，半马尔可夫条件随机场可以改写为以下的形式：

$$P(\boldsymbol{s}|\boldsymbol{x}; \boldsymbol{w}) = \frac{\exp(\boldsymbol{w} \cdot \Phi(\boldsymbol{x}, \boldsymbol{s}))}{\sum_{\boldsymbol{s}'} \exp(\boldsymbol{w} \cdot \Phi(\boldsymbol{x}, \boldsymbol{s}'))} \tag{7.4}$$

半马尔可夫条件随机场的训练过程即是对参数 w 的估计，那么最优参数 w^* 为：

$$\boldsymbol{w}^* = \arg\max_{w} \sum_{i=1}^{n} \log P(s^i|x^i; \boldsymbol{w}) \tag{7.5}$$

模型训练结束后，对给定的观测序列 $\boldsymbol{x}$，它对应的最优状态序列是：

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{s}} P(\boldsymbol{s}|\boldsymbol{x}; \boldsymbol{w}^*) \tag{7.6}$$

同条件随机场一样，可以使用维特比算法进行解码推断，得到最优状态序列 y^*。

2. 基于 Transformer 的命名实体识别

由于 Transformer 结构可以很好的并行化，并且具有较好的建模长文本的能力，因此基于 Transformer 结构的深度神经网络被广泛地应用于很多自然语言处理任务，如机器翻译，语言建模，预训练模型等。但是，直接在命名实体识别任务上使用 Transformer 模型往往表现不佳，其主要原因包括：

(1) **位置编码无法捕捉方向信息**：传统的 Transformer 使用正弦位置编码，这种编码可以捕捉距离信息，但是不能捕捉方向信息。例如，第 t 个字符的位置编码可以表示为：

$$PE_t = \begin{bmatrix} \sin(c_0 t) \\ \cos(c_0 t) \\ \vdots \\ \sin(c_{\frac{d}{2}-1} t) \\ \cos(c_{\frac{d}{2}-1} t) \end{bmatrix} \tag{7.7}$$

其中，d 表示位置编码的维度。根据余弦和差公式，对于距离第 t 个字符偏移量为 k 的字符，两个位置编码的点积可以写作：

$$PE_t^T PE_{t+k} = \sum_{j=0}^{\frac{d}{2}-1} [\sin(c_j t)\sin(c_j(t+k)) + \cos(c_j t)\cos(c_j(t+k))] = \sum_{j=0}^{\frac{d}{2}-1} \cos(c_j k) \tag{7.8}$$

这个点积反映了两个字符之间的距离。令 $j = t - k$，有：

$$PE_t^T PE_{t+k} = PE_j^T PE_{j+k} = PE_{t-k}^T PE_t. \tag{7.9}$$

从上面的式子可以看出，对于字符 t 偏移量为 $+k$ 和 $-k$ 的字符，结果是相同的，这意味着正弦位置编码不能捕捉到方向信息。然而，在命名实体识别任务中，词与词之间的相对位置是会影响模型最终判断的。例如，“复旦大学（ORG）位于上海” 和 “光华楼位于复旦大学（LOC）” 两个句子中，“复旦大学” 和 “位于” 处在不同的相对位置可以推断 “复旦大学” 分别为组织实体和地点实体。

(2) **平滑的注意力分布**：Transformer 模型在计算注意力时会对注意力分数进行缩放，得到一个

较为平滑的注意力分布。如公式7.10所示：

$$\text{Attn}(K, Q, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}})V \tag{7.10}$$

其中，d_k 是向量的维度，缩放因子 $\frac{1}{\sqrt{d_k}}$ 的目的防止较大的维度导致点积的幅度会增大，从而将 softmax 函数推入梯度极小的区域。但是对命名实体识别任务来说，一个词上下文中少数的词就足够用来判断它的标签，平滑的注意力分布反而可能会引入更多的噪声。

针对以上问题，TENER[2] 提出了一种改进版的 Transformer，其模型结构如图7.3所示。

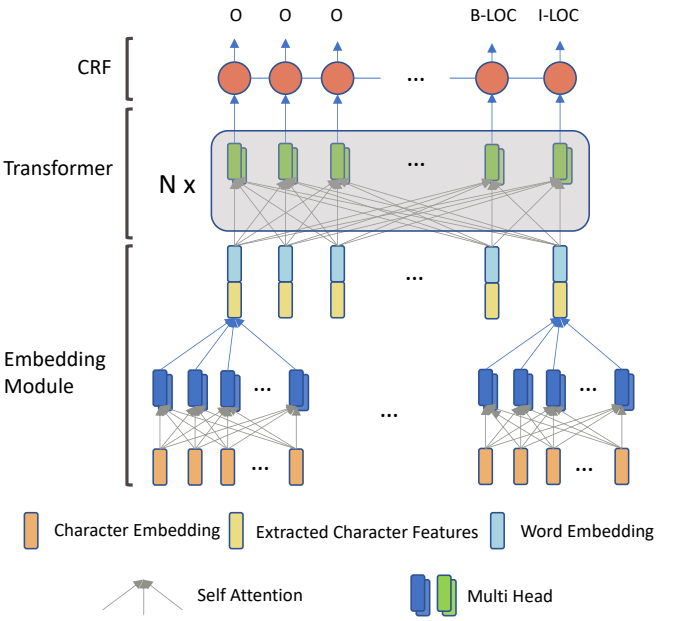


图 7.3 TENER 模型结构[2]

TENER 的主要有以下两点改进：(1) TENER 使用了相对位置编码，提升了方向的感知能力。新的相对位置编码为：

$$R_t, R_{-t} = \begin{bmatrix} \sin(c_0 t) \\ \cos(c_0 t) \\ \vdots \\ \sin(c_{\frac{d}{2}-1} t) \\ \cos(c_{\frac{d}{2}-1} t) \end{bmatrix}, \begin{bmatrix} -\sin(c_0 t) \\ \cos(c_0 t) \\ \vdots \\ -\sin(c_{\frac{d}{2}-1} t) \\ \cos(c_{\frac{d}{2}-1} t) \end{bmatrix}, \tag{7.11}$$

改进后，对于相同的位移 t，前向和后向的位置编码 R_t 和 R_{-t} 不完全相同。此时，注意力机制可以同时捕捉到距离信息和方向信息。(2) TENER 取消了计算注意力分数时的缩放因子，使得产生的注意力分数的分布更加稀疏:

$$\begin{aligned} R_{t-j} &= [\dots \sin(\frac{t-j}{10000^{2i/d_k}}) \cos(\frac{t-j}{10000^{2i/d_k}}) \dots]^T \\ A_{t,j}^{rel} &= Q_t^T K_j + Q_t^T R_{t-j} + K_j^T R_{j-t} + \boldsymbol{u}^T K_j + \boldsymbol{v}^T R_{t-j} \\ \text{Attn}(Q,K,V) &= \text{softmax}(A^{rel})V \end{aligned} \tag{7.12}$$

通常情况下，一句话只有少数实体，且仅需知道较小部分的上下文就可以判别实体的类别，而不需要关注所有词。因此，这种稀疏的注意力分数分布更适用于命名实体识别任务。

TENER 通过改进 Transformer 的位置编码及自注意力分数计算部分，使得 Transformer 也适用于命名实体识别任务。

3. 融合词典知识的栅格网络中文命名实体识别算法

中文命名实体识别与英文命名实体识别相比，由于中文单词边界不确定并且包含复杂的词语组合，使得中文命名实体识别更加困难。中文命名实体可以在中文分词基础上，利用序列标注技术对命名实体进行预测。也可以以中文汉字为基本单元构建序列标注任务。两者各有利弊，在中文分词结果基础上可以有效利用词汇信息，但是分词错误会累计传递。并且命名实体通常是未登录词，实体词在中文分词阶段可能被切分或者与前后词语错误组合，导致命名实体无法识别。而直接采用汉字作为基本单元，虽然可以避免中文分词的错误传递，但是无法利用词汇信息。比如：“南京市长江大桥”，如果没有词汇信息，识别结果很可能为：“南京”是地名，“江大桥”是人名。因此在基于汉字的中文命名实体识别模型上，如何使用词典信息来挖掘词语的特征成为中文命名实体识别研究者关注的焦点。

Zhang 等人提出了栅格结构长短时记忆网络（Lattice LSTM）[3] 来编码句子可以匹配所有词语信息。使用词典匹配句子中的单词，可以获得如图7.4所示的栅格（Lattice）的结构。栅格结构看作是一个有向无环图，词汇的开始和结束字符决定了其位置。对于输入汉字序列 $c_1, c_2, \dots c_n$，使用字典 $\mathbb{D}$ 匹配可以得到以 b 开始，以 e 结尾的子序列 $w_{b,e}^d$。上例中 $w_{1,2}^d$ 表示“重庆”，$w_{3,6}^d$ 表示“人和药店”。

基于汉字序列的 LSTM 模型中包含四种类型的向量：输入向量（Input Vectors）、隐藏向量（Output Hidden Vectors）、单元向量（Cell Vectors）和门向量（Gate Vectors）。输入向量 $\boldsymbol{x}_j^c = \boldsymbol{e}^c(c_j)$，表示输入汉字 c_j 的向量表示。$\boldsymbol{c}_j^c$ 表示单元向量，$\boldsymbol{h}_j^c$ 表示隐藏向量。原始 LSTM 门向量包含三个：输入门 $\boldsymbol{i}_j^c$、遗忘门 $\boldsymbol{f}_j^c$ 和输出门 $\boldsymbol{o}_j^c$。具体的计算公式见第 2 章公式 2.5 至公式 2.10。

Lattice LSTM 在原始 LSTM 的基础上，引入了词汇单元（Word Cell）$\boldsymbol{c}_{b,e}^w$ 表示从句子开始到当前 $w_{b,e}^w$ 的信息，如图7.5所示。词汇单元融合以该字符结束的所有词汇信息，例如图中“店”融合了“人和药店”和“药店”的信息。单元向量 $\boldsymbol{c}_j^c$ 不仅要考虑字符信息，还要考虑子序列 $w_{b,e}^w$ 的信息。

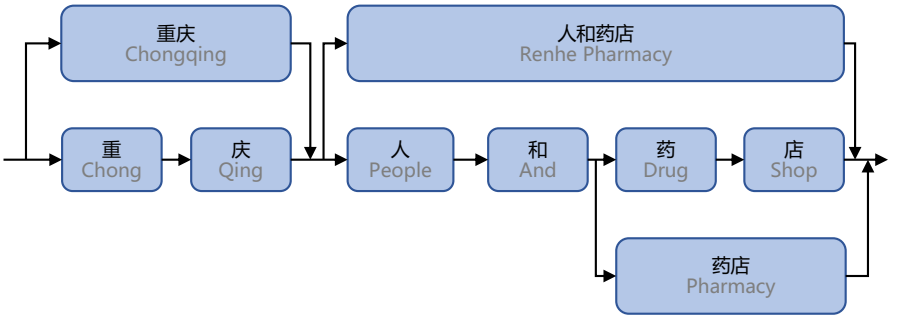


图 7.4 融合词典知识的栅格结构[3]

使用 $\boldsymbol{x}_{b,e}^{w} = \boldsymbol{e}^{w}(w_{b,e}^{s})$ 表示子序列 $w_{b,e}^{w}$ 的向量表示。词汇单元 $\boldsymbol{c}_{b,e}^{w}$ 具体计算公式如下：

$$\begin{bmatrix} \boldsymbol{i}_{b,e}^{w} \\ \boldsymbol{f}_{b,e}^{w} \\ \tilde{\boldsymbol{c}}_{b,e}^{w} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ tanh \end{bmatrix} \left(\boldsymbol{W}^{w\top} \begin{bmatrix} \boldsymbol{x}_{b,e}^{w} \\ \boldsymbol{h}_{b}^{c} \end{bmatrix} + \boldsymbol{b}^{w} \right) \tag{7.13}$$

$$\boldsymbol{c}_{b,e}^{w} = \boldsymbol{f}_{b,e}^{w} \odot \boldsymbol{c}_{b}^{c} + \boldsymbol{i}_{b,e}^{w} \odot \tilde{\boldsymbol{c}}_{b,e}^{w}$$

其中，$\boldsymbol{i}_{b,e}^{w}$ 和 $\boldsymbol{f}_{b,e}^{w}$ 分别是输入门和遗忘门。对于词汇单元没有输出门。

由于加入了词汇单元 $\boldsymbol{c}_{b,e}^{w}$，在计算当前字符的单元向量时 $\boldsymbol{c}_{j}^{c}$，会有多条路径的信息流。例如，对于“店”的单元状态（cell state）计算，即不仅包含它本身的信息（“店”字本身），还有对应匹配的词典信息（“人和药店”，“药店”），这里引入一个额外的门控单元 $\boldsymbol{i}_{b,e}^{c}$ 来控制每个词汇 $\boldsymbol{c}_{b,e}^{w}$ 的权重：

$$\boldsymbol{i}_{b,e}^{c} = \sigma \left(\boldsymbol{W}^{l\top} \begin{bmatrix} \boldsymbol{x}_{e}^{c} \\ \boldsymbol{c}_{b,e}^{w} \end{bmatrix} + \boldsymbol{b}^{l} \right) \tag{7.14}$$

在此基础上再引入注意力机制，计算得到当前字符的单元状态：

$$\boldsymbol{c}_{j}^{c} = \sum_{b \in \{b' | w_{b',j}^{d} \in \mathbb{D}\}} \boldsymbol{\alpha}_{b,j}^{c} \odot \boldsymbol{c}_{b,j}^{w} + \boldsymbol{\alpha}_{j}^{c} \odot \tilde{\boldsymbol{c}}_{j}^{c} \tag{7.15}$$

其中 $\boldsymbol{\alpha}_{b,j}^{c}$ 和 $\boldsymbol{\alpha}_{j}^{c}$ 由 $\boldsymbol{i}_{b,j}^{c}$ 和 $\boldsymbol{i}_{j}^{c}$ 归一化得到：

$$\boldsymbol{\alpha}_{b,j}^{c} = \frac{\exp(\boldsymbol{i}_{b,j}^{c})}{\exp(\boldsymbol{i}_{j}^{c}) + \sum_{b' \in \{b'' | w_{b'',j}^{d} \in \mathbb{D}\}} \exp(\boldsymbol{i}_{b',j}^{c})} \tag{7.16}$$

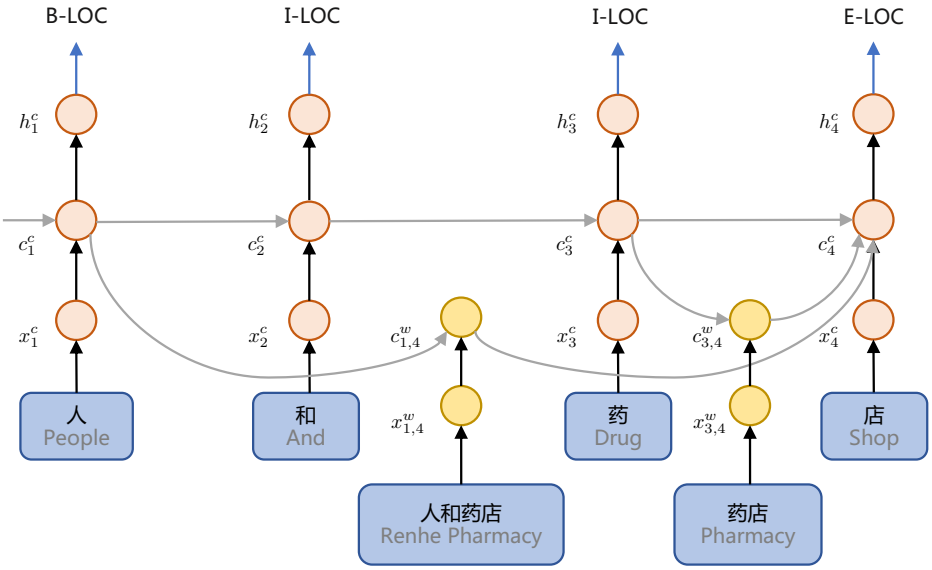


图 7.5 引入词汇单元的栅格结构长短时记忆网络结构图[3]

$$\boldsymbol{\alpha}_j^c = \frac{\exp(\boldsymbol{i}_j^c)}{\exp(\boldsymbol{i}_j^c) + \sum_{b' \in \{b'' | w_{b'',j}^d \in \mathbb{D}\}} \exp(\boldsymbol{i}_{b',j}^c)} \tag{7.17}$$

最后，可以在隐藏层 $\boldsymbol{h}_1^c, \boldsymbol{h}_2^c, ..., \boldsymbol{h}_n^c$ 之上添加标准 CRF 层得到最终输出，相关公式可以参考第 2 章公式 2.11、公式 2.12 和公式 2.13。

7.2.2 嵌套命名实体识别

嵌套命名实体（Nested Named Entity）是指在实体的内部还存在一个或多个其他的实体特殊实体类型。比如 “北京大学” 属于组织机构名实体，同时其中的 “北京” 又是地名类型的实体；“华为 P50 Pro” 属于产品类型的实体，其中 “华为” 又是公司名类型实体。嵌套命名实体在机构名、生物名词、化学名词等类型中普遍存在，相较于非嵌套识别难度更大。

嵌套命名实体识别问题可以形式化表示为：给定一个序列 $\boldsymbol{X} = \{x_1, x_2, \cdots, x_n\}$，其中 x_i 表示序列的第 i 个词或字，该序列对应的标签为 $\boldsymbol{Y} = \{y_1, y_2, \cdots, y_n\}$，其中 $y_i = \{y_i^1, y_i^2, \cdots, y_i^m\}$，$m$ 为标签嵌套层数。可以看到标签 y_i 与非嵌套命名实体识别不同，嵌套命名实体识别的标签 y_i 是包含多个标签的集合。HMM、CRF 等传统的序列标注方法只能对每个位置输出一个标签，无法解决多标签问题，因此不能直接应用于嵌套命名实体识别任务。此外嵌套命名实体的标签之间可能存在依赖关系，其嵌套的数量也不确定，这都是嵌套命名实体识别需要解决的难点。

解决嵌套命名实体识别的基本方法可以采用基于非嵌套实体识别算法的穷举法[4]，将所有可

能的单词序列都利用非嵌套实体识别算法进行识别和分类; 还可以将原有标签修改为组合形式, 将可能共同出现的所有类别进行组合, 产生新的标签体系（如: 将 B-Loc 与 B-Org 组合构造 B-Loc|Org 新标签）; 也才艺修改原有序列标注算法从单一目标到多目标, 利用 KL 散度等做为损失函数进行参数训练等方法。这些基本方法虽然实现相对简单并且直接, 但是存在计算消耗、标签量指数增加或者目标学习难度大等问题。本节将介绍几种实现相对复杂, 但是结果较好的算法, 包括: 基于成分句法分析、基于跨度以及基于生成式框架的嵌套式命名实体识别算法。

1. 基于成分句法分析嵌套命名实体识别

在本书第 3 章中, 我们介绍了成分句法理论, 知道了句法范畴之间不是完全对等的, 具有一定的层级关系。而嵌套命名实体中也存在着这样的层级关系, 因此不仅每个句子可以通过基于上下文无关文法的成分句法分析转化成其对应的成分语法树, 该句子中所包含的任意嵌套命名实体也可以转化成特定的树形结构。对比图7.6中句子的树形结构不难看出, 嵌套命名实体识别任务可以类比成分句法分析任务, 参考成分句法分析方法来解析句子进而实现句子中嵌套命名实体的识别。

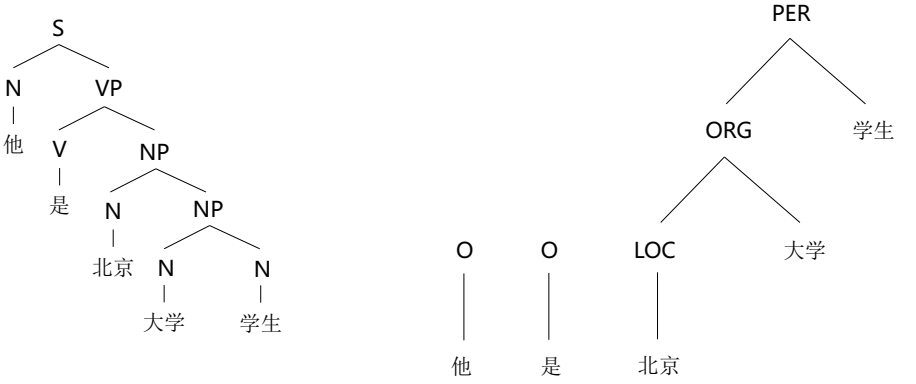


图 7.6 句子“他是北京大学学生”的成分语法树和嵌套命名实体分析

Finkel 和 Manning 提出了使用句法分析算法进行嵌套命名实体识别的思路[5], 基于此, 本节介绍一种基于状态转移的嵌套命名实体识别的方法, 其基本思想与移进-规约成分句法分析类似, 从左到右扫描输入的句子, 并维护三个部分: 堆栈 S、动作序列 A、队列 Q。其中, 堆栈 S 自底向上存储已规约的实体和待规约的单词, 动作序列 A 存储规约动作的历史信息, 队列 Q 存储尚未规约的单词。嵌套实体识别的移进-规约过程包含三种可能的动作:

- **移进（Shift）**: 将非空队列 Q 最左端的单词移入堆栈 S 中
- **规约-X（Reduce-X）**: 弹出堆栈 S 的两个元素 s_0 和 s_1, 将它们合并标注为实体类型 X 后重新压入堆栈 S 中

- 一元化-X（Unary-X）：弹出堆栈 S 的栈顶元素 s_0，将它标注为实体类型 X 后重新压入 S 中

图7.7展示了该方法在处理嵌套命名实体识别任务的流程，其中的 $ 作为句子的结束符被添加在所有句子的末尾。堆栈 S、动作序列 A、队列 Q 用于表示当前系统的状态，算法通过其当前状态来判断接下来应该执行的动作，由此实现状态的转移，从而完成识别过程。由于动作只有上面提到的三种，因此可以使用与句法分析相同的框架，将该问题转换为分类问题，输入为当前的系统状态 [S,A,Q]，输出为下一步的动作。通过正确的动作序列完成嵌套命名实体的识别。

受到句法树生成通常需要引入词性信息的启发，该方法同样将词性标注的信息引入词语的表示中：

$$\boldsymbol{e}_{x_i} = [\boldsymbol{e}_{w_i}, \boldsymbol{e}_{p_i}] \tag{7.18}$$

其中，$\boldsymbol{e}_{w_i}$ 是第 i 个词的词嵌入，$\boldsymbol{e}_{p_i}$ 是第 i 个词的词性标注嵌入。接着，该模型用两个长短期记忆网络（LSTM）学习任意时刻 k 下的历史动作序列 $A = \{a_0, a_1, ..., a_{k-1}\}$ 和队列 $Q = \{x_i, x_{i+1}, ..., x_n\}$ 的表示：

$$\begin{aligned} \boldsymbol{A}_k &= \overleftarrow{LSTM_a}[\boldsymbol{e}_{a_0}, ..., \boldsymbol{e}_{a_{k-1}}] \\ \boldsymbol{Q}_k &= \overrightarrow{LSTM_q}[\boldsymbol{e}_{x_i}, ..., \boldsymbol{e}_{x_n}] \end{aligned} \tag{7.19}$$

由于堆栈 S 中存放的是树形结构，因此可以使用在第 3 章第**??**节中所介绍的堆栈长短时记忆网络（Stack-LSTM）进行表示：

$$\boldsymbol{S}_k = \text{Stack-LSTM}[\boldsymbol{h}_{t_m}, ..., \boldsymbol{h}_{t_0}] \tag{7.20}$$

其中 h_{t_i} 表示从栈顶开始往下数第 i 个树形元素。该树形元素的非叶结点是用循环神经网络（Recursive Neural Network）按照如下公式计算得到：

$$\boldsymbol{h}_{parent} = \boldsymbol{W}_{u,l}^T \boldsymbol{h}_{child} + b_{u,l} \tag{7.21}$$

$$\boldsymbol{h}_{parent} = \boldsymbol{W}_{b,l}^T [\boldsymbol{h}_{lchild}, \boldsymbol{h}_{rchild}] + b_{b,l} \tag{7.22}$$

其中 $W_{u,l}$ 和 $W_{b,l}$ 分别是父节点标签为 l 时，一元操作 (u) 和二元操作 (b) 对应的权重矩阵，b 则是相应的正则化项。其叶子结点计算公式为：

$$\boldsymbol{h}_{leaf} = \boldsymbol{W}_{leaf}^T [\boldsymbol{e}_{x_i}, b_k] + b_{leaf} \tag{7.23}$$

至此，可以将三个部分的表示进行拼接得到任意时刻 k 下的系统状态 P_k 的表示：

$$\boldsymbol{P}_k = [\boldsymbol{S}_k, \boldsymbol{A}_k, \boldsymbol{Q}_k] \tag{7.24}$$

最后，可以利用已经训练好分类器来判断在系统状态 $\boldsymbol{P}_k$ 下应该执行的下一步动作，模型就

堆栈	动作	队列
Ø	Shift	他 是 北京 大学 学生 $
他	Shift	是 北京 大学 学生 $
他 是	Shift	北京 大学 学生 $
他 是 北京	Unary—LOC	大学 学生 $
LOC | 北京	Shift	大学 学生 $
LOC 大学 | 他 是 北京	Reduce—ORG	学生 $
ORG LOC 大学 | 他 是 北京	Shift	学生 $
ORG 学生 LOC 大学 | 他 是 北京	Reduce—PER	$
PER ORG 学生 LOC 大学 | 他 是 北京	Shift	$
PER $ ORG 学生 LOC 大学 | 他 是 北京	Ø	Ø

图 7.7 基于成分句法分析方法识别嵌套命名实体过程实例

具备了识别嵌套命名实体的能力。在训练分类器过程中，其损失采用所有训练数据中的动作损失之和：

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_{i}\sum_{k}\log p(z_{ik}) + \frac{\lambda}{2}||\boldsymbol{\theta}||^2 \tag{7.25}$$

其中 z_{ik} 是第 i 个句子的第 k 个动作，后半部分是 L2 正则项。

2. 基于跨度的嵌套命名实体识别

基于跨度的嵌套命名实体识别（Span-based Nested Named Entity Recognition）方法通过对句子的子序列进行分类来识别嵌套实体[6]。给定一个句子 $\boldsymbol{X} = \{x_1, x_2, \cdots, x_n\}$，其中 x_i 表示句子的第 i 个词或字，该句子中最大包含 k 个词或字的所有连续子序列 $[x_i, x_{i+1}, ..., x_j]$ 都会进行判定是否为某类实体。基于跨度的方法可以在一定程度上解决基于状态转移方法存在错误传播问题，但是传统的基于跨度的方法通常需要对所有子序列进行分类识别，因此推理效率很低。

针对缺乏跨度边界监督以及推理效率低等问题，Tan 等人提出了边界增强的基于跨度的嵌套命名实体识别算法 BENSC[7]。该算法将边界检测结合到跨度分类中，通过边界监督信号学习更好的表示，生成高质量的候选子序列来降低时间复杂度。模型总体结构如图7.8所示，主要包含边界检测和跨度分类两部分。边界检测旨在预测一个词是实体的第一个词还是最后一个词。跨度分类旨在将可能的跨度分类为相应的语义标签。两部分在多任务学习框架下联合训练。

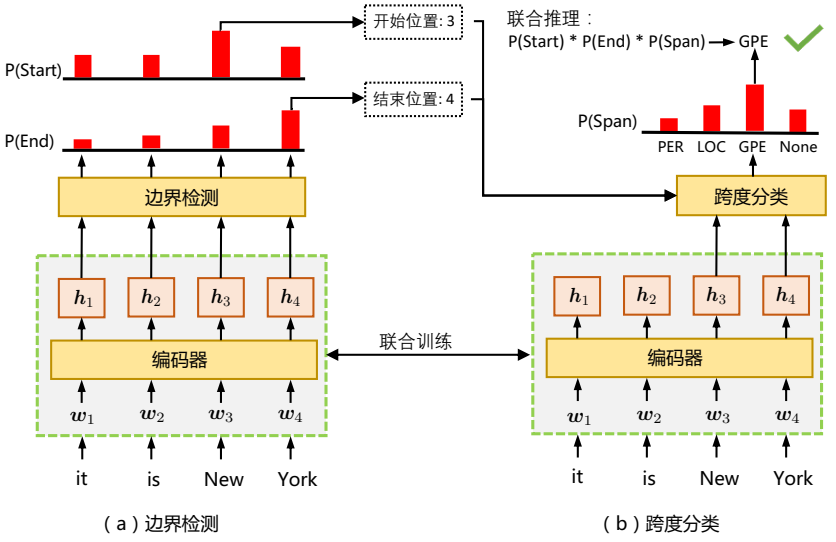


图 7.8　边界增强的基于跨度的嵌套命名实体识别模型结构图[7]

具体来说，通过编码器将编码后将文本上下文表示 $\boldsymbol{h}_i$ 输入到多层感知器（Multilayer Perceptron，

MLP）分类器中，并应用 Softmax 函数来获得单词 w_i 作为实体的第一个单词的概率 P_s^i：

$$P_s^i = \text{Softmax}(MLP_{start}(\boldsymbol{h}_i)) \tag{7.26}$$

同样，可以应用 MLP 分类器来获得单词 w_i 是实体的最后一个单词的概率 P_e^i：

$$P_e^i = \text{softmax}(MLP_{end}(\boldsymbol{h}_i)) \tag{7.27}$$

在训练过程中,由于每个句子可能包含多个实体,于是将所有实体的跨度边界标记为正确答案。将边界检测任务的训练目标函数定义为以下两个交叉熵损失的总和，分别检测开始和结束边界：

$$\begin{aligned}
\mathcal{L}_{bdr}^s &= -\sum_{i=1}^{N}[y_s^i \log P_s^i + (1-y_s^i)\log(1-P_s^i)] \\
\mathcal{L}_{bdr}^e &= -\sum_{i=1}^{N}[y_e^i \log P_e^i + (1-y_e^i)\log(1-P_e^i)] \\
\mathcal{L}_{bdr} &= \mathcal{L}_{bdr}^s + \mathcal{L}_{bdr}^e
\end{aligned} \tag{7.28}$$

其中 y_s^i 和 y_e^i 分别表示词 i 是否是实体的第一个或者最后一个词。

根据边界检测得到的可能实体的第一个词和最后一个词，使用编码器得到的跨度表示 $\boldsymbol{v}_{sp}$ 输入到全连接层 MLP 中进行分类：

$$P_{sp} = \text{softmax}(MLP_{sp}(\boldsymbol{v}_{sp})) \tag{7.29}$$

k 表示语义标签的数量，y_{sp}^t 表示跨度 $(w_i, ..., w_j)$ 是否是标签 t，最小化如下的交叉熵损失函数：

$$\mathcal{L}_{sp} = -\sum_{t=1}^{k}(y_{sp}^t \log P_{sp}^t + (1-y_{sp}^t)\log(1-P_{sp}^t)) \tag{7.30}$$

总体联合训练的损失函数如下:

$$\mathcal{L} = w\mathcal{L}_{bdr} + (1-w)\mathcal{L}_{sp} \tag{7.31}$$

其中 w 是平衡边界检测和跨度分类的超参数。

模型在推断过程中，对于给定的实例 $(w_i, ..., w_j)$，首先从边界检测模型中获得开始和结束的边界概率 P_s^i 和 P_e^i。将 $j > i$ 且 $P_s^i * P_e^i$ 大于预先设定的阈值的区域或子序列作为合法的跨度。最后将合法的跨度输入到跨度分类中得到分类结果。

7.2.3 多规范命名实体识别

前面两节分别讲述了非嵌套和嵌套的命名实体识别方法，这些方法只能处理特定标注范式的实体，方法之间不具备普适性及迁移性。但实际应用时可能会遇到多种范式的实体，包括非嵌套命名实体（Flat Named Entity）、嵌套命名实体（Nested Named Entity）、不连续命名实体（Discontinuous Named Entity）等不同的子任务。当前模型需要分别采取序列标注、基于跨度和基于转移的方法进行处理，这些解决方案很难同时处理上述所有子任务。为了使得命名实体识别任务在实际应用中取得更好的效果，需要一个能同时处理不同实体范式的统一命名实体识别方法。

Yan 等人[8] 提出了一个面向命名实体识别不同子任务的序列到序列的生成框架。该方法使用指针方式，将序列标注任务转化为序列生成任务，使用序列到序列（Seq2Seq）的范式来进行生成，将预训练的序列到序列模型 BART 融入框架，并利用三种实体表示将实体线性化为一个序列，这样就可以通过统一的序列到序列模型来完成上述三个子任务。

首先，以生成式的序列对不同类型的实体进行统一表示。当给定由 n 个词组成的输入语句 $\boldsymbol{X} = [x_1, x_2, ..., x_n]$ 和，目标序列 $\boldsymbol{Y} = [s_{11}, e_{11}, ..., s_{1j}, e_{1j}, t_1, ..., s_{i1}, e_{i1}, ..., s_{ik}, e_{ik}, t_i]$，其中 s, e 分别是一个实体跨度的起止索引，对于非嵌套和嵌套命名实体识别任务，每个实体只包含一个跨度，但是对于不连续 NER 任务，每个实体包含多个跨度，因此每个实体表示为 $[s_{i1}, e_{i1}, ..., s_{ik}, e_{ik}, t_i]$，其中 t_i 是实体标签索引。定义 $\boldsymbol{G} = [g_1, g_2, ..., g_l]$ 为实体标签列表，l 是实体标签数量，为了和指针索引区分，对 t_i 做一个长度为 n 的偏移，即 $t_i \in \{n+1, ..., n+l\}$。

例如：　输入：北京大学

输出：1 2 5 1 4 6

其中，输出序列中 1 和 2 表示第 1 个实体跨度的起止索引，5 表示 LOC 实体类型，接下来输出序列中的 1 和 4 表示第 2 个实体跨度的起止索引，6 表示 ORG 实体类型。

根据上述定义，可以通过对目标序列 $\boldsymbol{Y}$ 的条件概率进行建模得到句子中的实体表示。

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{t=1}^{m} P(y_t|\boldsymbol{X}, \boldsymbol{Y}_{<t}) \tag{7.32}$$

针对条件概率最大化问题，可以采用具有生成能力的 BART 模型求解。模型主要包含两部分：编码器和解码器，模型结构如图7.9所示。

编码器将输入的语句 $\boldsymbol{X}$ 编码为词嵌入表示：

$$\boldsymbol{E}^e = \text{TokenEmbed}(\boldsymbol{X}) \tag{7.33}$$

然后通过特征抽取，得到特征向量表示 $\boldsymbol{H}^e$：

$$\boldsymbol{H}^e = \text{Encoder}(\boldsymbol{X}) \tag{7.34}$$

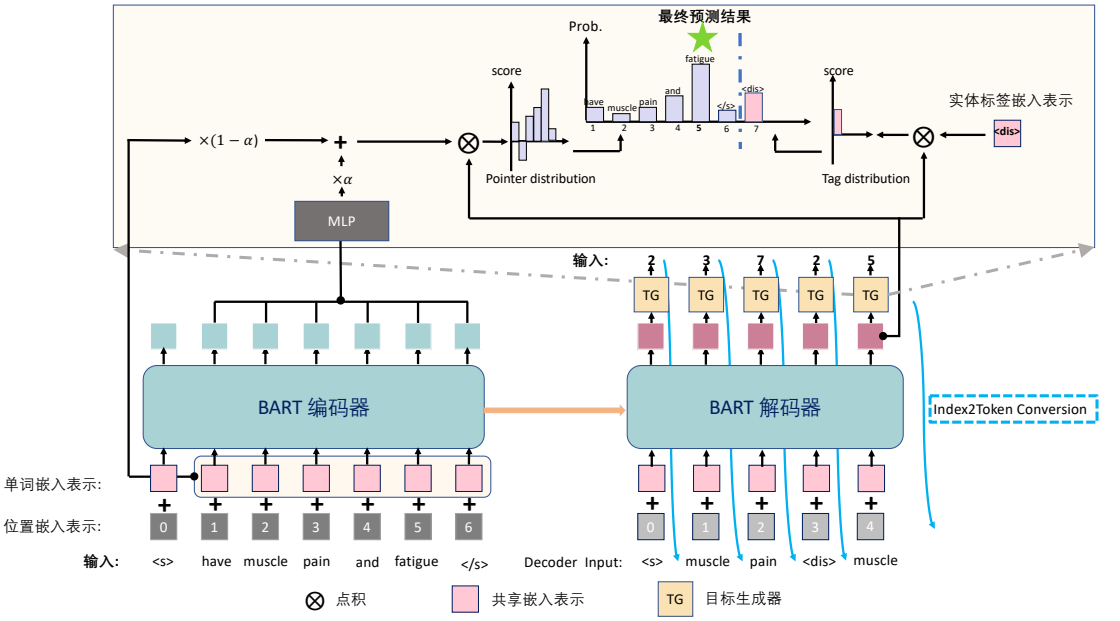


图 7.9 基于序列到序列的多规范命名实体识别方法的模型结构[8]

在通过一个多层感知器后，将得到的结果 $\hat{\boldsymbol{H}}^e$ 与词嵌入表示 $\boldsymbol{E}^e$ 按不同权重相加：

$$\overline{\boldsymbol{H}}^e = \alpha * \hat{\boldsymbol{H}}^e + (1-\alpha) * \boldsymbol{E}^e \tag{7.35}$$

其中 α 是超参数。

解码器在每个时刻计算索引的概率分布 $P_t = P(y_t|\boldsymbol{X}, \boldsymbol{Y}_{<t})$，输出为指针索引或者实体标签索引：

$$\hat{y}_t = \begin{cases} X_{y_t} & if \quad y_t \leqslant n \\ G_{y_t - n} & if \quad y_t > n, \end{cases} \tag{7.36}$$

然后将编码器的输出和索引表示输入到解码器：

$$h_t^d = \text{Decoder}(\boldsymbol{H}^e; \hat{\boldsymbol{Y}}_{<t}) \tag{7.37}$$

用如下公式计算出索引的概率分布：

$$P_t = \text{softmax}([\overline{\boldsymbol{H}}^e \otimes h_t^d; G^d \otimes h_t^d]) \tag{7.38}$$

其中

$$G^d = \text{TokenEmbed}(G) \tag{7.39}$$

最终，使用算法7.1可以将目标索引序列将其解析成抽取出的实体及类型。

代码 7.1: 将实体表示序列转换成实体跨度

输入: 目标序列 $Y = [y_1, \ldots, y_m], y_i \in [1, n + |G|]$
输出: 实体跨度 $E = (e_1, t_1), \ldots, (e_i, t_i)$

```
E = , e = [], i = 1
while i <= m do
    y_i = Y[i]
    if y_i > n then
        if len(e) > 0 then
            E.add((e, G_{y_i - n}))
        end if
        e = []
    else
        e.append(y_i)
    end if
    i = i + 1
end while
return E
```

值得注意的是，BART 使用的字节对编码（Byte-Pair-Encoding，BPE）编码方式会将一个单词处理成若干子词，所以 BART 框架在使用时需要进行适当的修改，可以采用如下三种基于指针的实体表示来明确地定位原句子中的实体：

- Span：实体的每个起始位置与结束位置。
- BPE：实体中每个词的所有 BPE 对应的位置索引。
- Word：每个实体的字的第一个 BPE 对应的位置索引。

以图7.10所示的句子为例，句子中有三个实体，(x_1,x_3,PER)，(x_1, x_2, x_3, x_4,LOC)，(x_4,ORG)，其中 PER、LOC 和 ORG 是实体类型。利用上述三种表示方法，可以得到如下表示：

- Span：[0,2,5,5,PER]，[0,7,LOC]，[6,7,ORG]
- BPE：[0,1,2,5,PER]，[0,1,2,3,4,5,6,7,LOC]，[6,7,ORG]
- Word：[0,5,PER]，[0,3,5,6,LOC]，[6,ORG]

在大多数情况下，使用 Word 实体表示的结果会更好。与 Span 实体表示相比，由于 BPE 实体表示与预训练任务更相似，所以效果比 Span 实体表示效果更好。但是当数据集中一个实体被标记

成很长的 BPE 序列的时候，Span 实体表示的效果会优于其他两种。

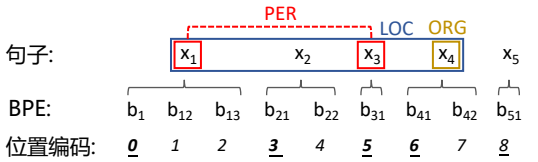


图 7.10　三种基于指针的实体表示方法

7.2.4 命名实体识别评价方法

一般意义上，命名实体识别任务同时涉及两种识别目标，即实体的边界和实体的类型。相应地，对于精确匹配评估而言，实体识别任务的成功标准为，实体边界以及实体类别同时被精确地标出。依据预测结果真阳性（True Positives，TP），假阳性（False Positives，FP），及假阴性（False Negatives，FN），能够得出 NER 任务的精确率，召回率以及 F-score 用于评估任务优劣。对 NER 中的 TP，FP 与 FN 有如下解释：

TP：NER 能正确识别实体。

FP：NER 识别实体的类型或边界错误。

FN：应该但没有被 NER 所识别的实体。

精度（Precision，P）评价的是 NER 模型预测的实体中正确的占比，召回（Recall，R）评价的是 NER 模型预测出正确的实体在整个数据集中的占比，它们的计算公式如下：

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}. \tag{7.40}$$

F-值通常定义为精度以及召回的调和平均，其中最为常用的 F 值是：

$$\text{F1} = 2 \times \frac{\text{P} \times \text{R}}{\text{P} + \text{R}}. \tag{7.41}$$

绝大多数的 NER 任务涉及对多种实体类别进行识别，这就要求对所有的实体类别评估 NER 的效果。为了实现这一目的，与词性标注问题相同，通常借助两类评估指标，即宏平均 F1（Macro-F1）和微平均 F1（Micro-F1）。宏平均 F1 值将所有的实体类别都视为平等的，先在单个实体类别上计算其 F1 值，继而求得整体的平均值。微平均 F1 值将每个实体个体都视为平等的，直接对整体数据求得 F1 值。宏平均 F1 对含有较少数量实体类别类别更敏感。

7.2.5 命名实体识别语料库

命名实体识别任务与自然语言处理其他任务类似，目前也大都采用有监督方法进行建模，因此需要大规模的标注数据进行模型训练和对比评测。表7.2给出了常用的非嵌套命名实体识别语料

库汇总，并根据语料库中文本语言、文本类型以及实体类型标签数以及是否包含嵌套实体等不同可以进行划分。

表 7.2 命名实体识别语料库汇总

语料库	语言	文本类型	标签数	嵌套实体比例
CoNLL2003	英文、德文	新闻	4	0%
OntoNotes	英文、中文、阿拉伯文	新闻、广播、电话语音	11	0%
ACE 2004	英文、中文、阿拉伯文	新闻	7	17%
ACE 2005	英文、中文、阿拉伯文	新闻	7	17%
GENIA	英文	生物	4	30%
MSRA	中文	新闻	3	0%
Weibo NER	中文	社交媒体	4	0%
Resume	中文	简历	8	0%
CLUE Fine-Grain NER	中文	新闻	10	0%
CCKS2018 电子病例	中文	电子病历	5	0%

1. 非嵌套英文命名实体识别语料库

大多数的英文命名实体识别的工作主要是利用 CoNLL2003[9] 和 OntoNotes 两个数据集进行训练和评测。CoNLL2003 包括 1393 篇英语新闻和 909 篇德语新闻经过人工标注组成。英语新闻来源于 Reuters Corpus Volume 1（RCV1）。标注人员对文本内容中的实体进行了标注了，实体类型包括：人名，地名，组织名和其他实体。在数据集中还包含了每个单词的词性信息。

OntoNotes 迄今共为止发行五个版本，OntoNotes 5.0[10] 版本包含英文、中文以及阿拉伯文三种语言的新闻、广播、电话对话、博客、新闻组、脱口秀等类型的文本。该语料库中不仅包含命名实体标注，还包括语法结构、谓词论元结构等结构信息，还包含词义消歧、指代消解等语义信息。在命名实体标注方面，OntoNotes 5.0 中标注了 11 种命名实体（包括人名、地名、机构名等），还标注了 7 种类实体（包括日期、时间、百分比等）。OntoNotes 5.0 语料集规模较大，三种语言总计包含 290 万单词，仅英文的新闻语料就包含 62.5 万单词，为命名实体识别提供了很好的训练和测试基准。

2. 非嵌套中文命名实体识别语料库

中文命名实体识别工作基本评测语料库主要包括 MSRA[11]、OntoNotes[10]、Weibo NER[12] 等。MSRA 数据集合做为 SIGHAN 2003 评测中的一部分，除了中文分词之外还标注了命名实体，主要针对人名、地名、机构名三种类型实体。随着命名实体识别研究的不断深入，针对特定领域的命名实体识别语料库也不断涌现，包括医药、生物、社交媒体、电子病历等领域数据集也受到广泛关注。Weibo NER 数据集合中包含从新浪微博收集的 226 万无标注数据和 1890 条标注数据。

3. 嵌套命名实体识别语料库

除了上述介绍的中英文非嵌套命名实体识别数据集，常用的含有嵌套命名实体的数据集主要为新闻领域的数据集 ACE 2004[13]、ACE 2005[14], 以及生物医学领域的数据集 GENIA[15]。ACE 2004 和 ACE 2005 数据集中主要包含 7 种实体类型，其中含有嵌套命名实体的句子占 30% 左右。ACE 2004 数据集中英文数据大约 15.8 万词，中文 15.4 万词，阿拉伯文 15.1 万词的标注规模。GENIA 数据集中主要包含 4 种实体类型，其中含有嵌套命名实体的句子占 17% 左右。GENIA V3.0 语料集针对 2000 篇 MEDLINE 系统中 2000 篇论文的摘要共计 40 万单词进行了标注。

7.3 关系抽取

关系抽取（Relation Extraction，RE）最初是在 1998 年 MUC-7 会议上首次正式提出，旨在从无结构文本中识别两个或多个实体之间的语义关系，是信息检索、智能问答、人机对话等应用系统中不可获取的基础任务，也是知识图谱构建所依赖的关键技术之一。本节主要介绍二元关系抽取，关注两个实体之间的语义关系。实体间的关系可以用 <Head，Relation，Tail> 三元组进行表示，其中 Head 和 Tail 分别表示头实体和尾实体，Relation 表示实体之间的关系类型。

例如：根据句子“刘翔出生于上海”

可抽取 < 刘翔，出生地，上海 >，表示“刘翔”和“上海”之间存在“出生地”关系。

关系抽取任务的主要难度在于关系类型种类繁多以及对语义建模能力要求高。以 Freebase 知识库为例，其包含 4000 种关系类型和 7000 种属性类型[16]。如果考虑多实体关系以及关系之间的重叠，关系类型将更加复杂。面对如此庞大且不断增长的关系类型，目前大多数基于有监督方法的关系抽取任务通常根据应用的不同，构建特定领域的关系抽取模型，从而大幅度降低了模型的复杂程度。但是在处理不同领域任务时，需要重复进行关系类型定义、标注数据收集、模型训练等环节，这在一定程度上制约了关系抽取算法的通用性。此外，描述实体之间关系的语言丰富，形式也多种多样，这进一步增加了关系抽取任务的难度。

例如：(1)：复旦大学始创于 1905 年，位于中国上海。

(2)：复旦大学 地址：上海市杨浦区邯郸路 220 号

(3)：杨浦区域坐落着 14 所各类高等院校，包括：复旦大学、同济大学等

上述三个句子都表明了“复旦大学”和“上海”之间存在“位于”关系，但是其表达形式之间的差别却非常大，如何能够建模这种长距离、丰富内容且形式变化多样的语义关系是关系抽取算法迫切需要解决难题。

从关系抽取任务定义可以看到其目标是识别实体间语义关系，因此依据是否已经在无结构文本中标记了实体类型，关系抽取方法可分为**联合式抽取**和**流水线式抽取**。联合式抽取是指利用单个算法完成从文本中同时完成命名实体识别和关系抽取。流水线式抽取则是首先使用命名实体识别算法识别文本中的实体，然后构造关系抽取模型识别实体对间的关系。两种方法各有利弊，流水线式方法可以将复杂的关系抽取任务拆解，从而降低单个模型的复杂程度，但是会带来错误传

递的风险。而联合式抽取算法可以有效缓解错误传递的问题，但是模型相对复杂，文本中不仅包含有语义关联的实体对，也包含单纯的命名实体，这会造成模型学习难度的急速提升，影响模型学习效果。

根据关系类型是否需要提前预先定义，关系抽取算法可以分为**预定义关系抽取**和**开放关系抽取**两类。预定义关系抽取是指针对一个或者多个领域内预先定义的实体间关系进行抽取。开放关系抽取则针对不限定领域的范围和关系类别的抽取任务。预定义关系抽取算法还可以根据所使用的训练数据是标注数据还是外部知识自动标注的情况细分为基于有监督和远程监督等类型。本节中将分别针对有监督关系抽取、远程监督关系抽取以及开放关系抽取三类方法进行介绍。

7.3.1 有监督关系抽取

有监督的关系抽取方法将关系抽取问题转换为多分类问题。关系抽取任务的为输入文本内容和待判断的提及对（Mention Pair），输出为提及对之间根据所给定的文本内容所表达的关系类型。有监督的关系抽取方法需要在大量的人工标注训练语料，通过设计有效的特征来构建各类分类模型。也可以利用神经网络自动提出语义特征进行关系抽取。本节将介绍基于最大熵关系抽取算法和基于图卷积网络的关系抽取方法。

1. 基于最大熵的关系抽取

基于最大熵的统计建模方法可以很好综合地考虑文本中的词汇、句法、语法和语义特征，因此在自然语言处理有广泛的应用。最大熵模型是由最大熵原理推导得出的用于分类的对数线性学习模型。其基本原理是：对于一个未知分布，在只掌握部分信息的情况下，不能对未知信息引入任何主观的假设，同时应该充分利用已经掌握的已知信息。最大熵模型假设熵值最大的概率分布能够最真实地反映事件的分布情况。

熵度量了事件的不确定性，对于越不确定的事件，其熵值就越大。具体地，对于离散型随机变量 X，$P(x)$ 表示 X 的概率分布，X 的熵可以表示为：

$$H(X) = -\sum_{x} P(x) \log P(x) \tag{7.42}$$

对于离散型随机变量 Y，在已知随机变量 X 的条件下，条件熵 $H(Y|X)$ 表示为：

$$H(Y|X) = \sum_{i=1}^{n} P(x_i) H(Y|X = x_i) = -\sum_{x,y} P(x)P(y|x) \log P(y|x) \tag{7.43}$$

最大熵原理的目标是在所有满足约束条件的概率分布中，选择使熵值最大的概率分布。因此，基于最大熵原理的目标函数为：

$$\max H(Y|X) = -\sum_{x,y} P(x)P(y|x) \log P(y|x) \tag{7.44}$$

为了方便使用凸优化的方法，通常将最大值问题改写为最小值问题，即最终的目标函数为：

$$\min -H(Y|X) = \sum_{x,y} P(x)P(y|x)\log P(y|x) \tag{7.45}$$

给定一个具有 n 个样本的训练集 $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$，其中 x_i 为模型的输入特征，y_i 为 x_i 所对应的样本类别输出。定义 $m(X = x, Y = y)$ 为样本 (x, y) 出现的次数，随机变量 X 和 Y 的联合分布 $P(X, Y)$ 的经验分布 $\tilde{P}(X, Y)$ 以及 X 的边缘分布 $P(X)$ 的经验分布 $\tilde{P}(X)$ 可以被定义为：

$$\tilde{P}(X, Y) = \frac{m(X = x, Y = y)}{n} \tag{7.46}$$

$$\tilde{P}(X) = \frac{m(X = x)}{n} \tag{7.47}$$

对于训练集中存在的一些事实关系可以通过特征函数描述。同一个样本可以具有多个特征函数，并最终通过特征函数来约束模型。对于输入特征 x 和类别输出 y 的特征函数 $f(x, y)$：

$$f(x, y) = \begin{cases} 1, & x, y\text{满足某个条件} \\ 0, & \text{其他} \end{cases} \tag{7.48}$$

特征函数 $f(x, y)$ 关于经验分布 $\tilde{P}(X, Y)$ 的期望 $E_{\tilde{P}(f)}$ 为：

$$E_{\tilde{P}(f)} = \sum_{x,y} \tilde{P}(X, Y)f(x, y) \tag{7.49}$$

特征函数 $f(x, y)$ 关于条件分布 $P(Y|X)$ 和经验分布 $\tilde{P}(X)$ 的期望 $E_{P(f)}$ 可以表示为：

$$E_{P(f)} = \sum_{x,y} \tilde{P}(X)P(y|x)f(x, y) \tag{7.50}$$

假设训练集中有 M 个特征函数 $\{f_i(x, y)\}_{i\in[1,M]}$，分别对应最大熵模型的 M 个约束条件。模型从特征 f_i 中学习参数，上述两个数学期望值应相等，即：

$$E_{P(f)} = E_{\tilde{P}(f)} \tag{7.51}$$

最大熵模型的优化问题可以表示为：

$$\begin{aligned} \arg\min_{p\in P} -H(P) &= \sum_{x,y} \tilde{P}(x)P(y|x)\log P(y|x) \\ s.t. \quad & E_{P(f_i)} = E_{\tilde{P}(f_i)}, i = 1, \cdots, M \\ & \sum_{y} P(y|x) = 1 \end{aligned} \tag{7.52}$$

拉格朗日乘子法可以在满足约束条件下求解目标函数的最优解。可以证明，满足约束条件的最优解可以表示为：

$$\hat{P}(y|x) = \frac{1}{Z(x)} \exp(\sum_{i=1}^{M} \omega_i f_i(x,y)) \tag{7.53}$$

其中，

$$Z(x) = \sum_{Y} \exp(\sum_{i=1}^{M} \omega_i f_i(x,y)) \tag{7.54}$$

最大熵模型的训练过程可以简述为选取有效的特征 f_i 及其权重 ω_i。所以能够包含更多语义信息的特征条件对最大熵模型的训练尤为关键。

针对关系抽取问题，提及对之间所具备的关系需要通过输入的文本内容进行确定，因此关系抽取中通常需要抽取提及本身以及提及对之间的特征，常见的特征有：

单词特征：提及对中两个提及的单词和这两个提及中间的所有单词。

实体类型特征：两个提及所表示的实体的类型。

重叠性特征：两个提及中间的单词数目；两个提及中间其他提及的数量以及指示这两个提及是否是相同的词性（如名词或者动词）的标志。

依赖性特征：依存关系树中提及对所依赖的单词以及单词的词性和组块标签。

语法树特征：语法树中连接这提及对的非终结符 (去掉重复项) 的路径。

例如,“上市公司拒绝了曾担任其董事会主席-鲍勃的请求”中的句子片段“担任其董事会主席”所对应的语法树如图7.11所示。

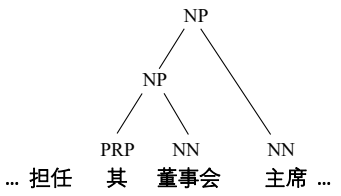


图 7.11　句子片段“担任其董事会主席”所对应的语法树结构图

对于该句子片段中的一个提及对（“董事会”和“主席”），可以从特征规则中抽取的特征如下所示：

单词特征：$PERSON_{m1}$（“主席”）,ORG_{m2}（“董事会”）。

实体类型特征：$NOMINAL_{m1}, NOMINAL_{m2}$。

重叠性特征：两个提及中间的单词数目 (0)。

语法树特征：语法树中连接这两个提及的非终结符的路径（PERSON-NP-ORGANIZATION）。

基于最大熵的关系抽取方法使用从上述特征规则中提取到的特征来训练最大熵模型。通过将文本中各式各样的丰富的语义信息整合到特征中，并配合最大熵模型，从而实现具有良好拓展性和表现的有监督关系抽取模型。

2. 基于图卷积网络的关系抽取

关系抽取需要根据句子抽取实体之间的关系，句子的句法结构提供了捕捉单词之间的长距离关系的有效信息。Zhang 等人[17] 提出了通过图卷积神经网络（Graph Convolutional Network, GCN）有效利用句子依存句法树结构的关系抽取算法。通过图卷积操作对输入句子的依存结构进行编码，然后提取以实体为中心的表示，从而进行关系预测。模型结构如图7.12所示。

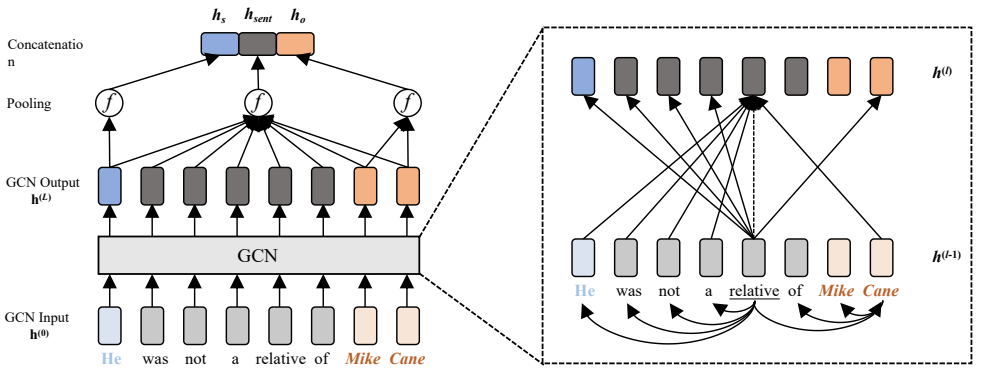


图 7.12　基于图卷积网络的关系提取模型结构[17]

给定一个拥有 n 个节点的图，可以使用一个 $n \times n$ 的邻接矩阵 $\boldsymbol{A}$ 来表示这个图结构。其中，如果节点 i 到节点 j 之间存在一条边的话，则有 $A_{ij} = 1$。在一个拥有 L 层的图卷积网络中，假定第 l 层中节点 i 的输入向量为 $\boldsymbol{h}_i^{(l-1)}$，输出向量为 $\boldsymbol{h}_i^{(l)}$，那么一个图卷积操作可以表示为：

$$\boldsymbol{h}_i^{(l)} = \sigma(\sum_{j=1}^{n} A_{ij} \boldsymbol{W}^{(l)} \boldsymbol{h}_j^{(l-1)} + b^{(l)}) \tag{7.55}$$

其中，$\boldsymbol{W}^{(l)}$ 是线性转换操作，$b^{(l)}$ 是偏置项，σ 是非线性函数（如 ReLU）。在每次图卷积的过程中，图中的每个节点会汇聚并总结来自于相邻节点的信息。

为了将图卷积操作适用于依存树，可以将每棵依存树转换成相应的邻接矩阵 $\boldsymbol{A}$。其中，如果单词 i 和 j 之间存在一条依存边的话，则有 $A_{ij}=1$。然而，因为单词变化很大，简单地应用公式7.55中的图卷积运算可能会导致节点表示的幅值差异很大。这可能会使句子表征偏向于高度节点，而忽略节点本身所携带的信息是什么。除此之外，因为节点在依存树中不会与自己相连接，所以 $\boldsymbol{h}_i^{(l-1)}$ 的信息将永远不会传递给 $\boldsymbol{h}_i^{(l)}$。

为了解决上述问题，Zhang 等人[17] 在向图卷积网络输入非线性信息之前对其隐激活进行标准化，并向图中的每个节点添加自循环，具体做法如下：

$$\boldsymbol{h}_i^{(l)}=\sigma(\sum_{j=1}^{n}\tilde{A}_{ij}\boldsymbol{W}^{(l)}\boldsymbol{h}_j^{(l-1)}/d_i+b^{(l)}) \tag{7.56}$$

其中，$\tilde{\boldsymbol{A}}=\boldsymbol{A}+\boldsymbol{A}$，$\boldsymbol{A}$ 是 $n\times n$ 的单位矩阵，这就相当于给每个节点加自循环，这样 $\boldsymbol{h}_i^{(l-1)}$ 的信息就可以传递给 $\boldsymbol{h}_i^{(l)}$ 了；$d_i=\sum_{j=1}^{n}\tilde{A}_{ij}$ 是生成图中词元 i 的度，将它放在分母项就可以对节点的度进行标准化，解决了节点表示幅值差异过大的问题。

把上述操作叠加在 L 层上就形成了一个深层的图卷积网络。其中，将 $\boldsymbol{h}_1^{(0)},...,\boldsymbol{h}_n^{(0)}$ 作为输入词向量，$\boldsymbol{h}_1^{(L)},...,\boldsymbol{h}_n^{(L)}$ 作为输出词标识特征。这样一来，神经网络中的所有操作就都可以用矩阵乘法实现，这样就非常适合在 GPU 上进行批量计算。并且，信息在单词间的传播是并行进行的，其运行时间就不受依赖树深度的影响。

$\boldsymbol{X}=[x_1,...,x_n]$ 代表一个句子，其中 x_i 是第 i 个单词。主语实体表示为 $\boldsymbol{X}_s=[x_{s_1},...,x_{s_2}]$，宾语实体表示为 $\boldsymbol{X}_o=[x_{o_1},...,x_{o_2}]$。假如预先给定一个关系集合 $\boldsymbol{R}$，并且已经给出 $\boldsymbol{X}$、$\boldsymbol{X}_s$、$\boldsymbol{X}_o$，关系抽取的目的就是预测实体间的关系 $r\in\boldsymbol{R}$ 或是“无关系”。

在词向量上应用了 L 层 GCN 之后，我们就可以获得每个单词的隐藏表示，这些单词在依存树中直接受其距离不超过 L 条边的邻居节点的影响。为了在关系抽取任务中利用这些词表示，我们首先得到一个句子表示，

$$\boldsymbol{h}_{\text{sent}}=f(\boldsymbol{h}^{(L)})=f(\mathbf{GCN}(\boldsymbol{h}^{(0)})) \tag{7.57}$$

其中，$\boldsymbol{h}^{(l)}$ 代表着 GCN 在第 l 层的集合隐藏表示；$f:\boldsymbol{R}^{d\times n}\rightarrow\boldsymbol{R}^d$ 是一个从 n 个输出向量映射到句子向量的最大池化函数。同样地，我们可以从 $\boldsymbol{h}^{(L)}$ 获取到主语和宾语的隐藏表示，如下所示：

$$h_s=f(\boldsymbol{h}_{s_1:s_2}^{(L)}) \tag{7.58}$$

$$h_o=f(\boldsymbol{h}_{o_1:o_2}^{(L)}) \tag{7.59}$$

通过连接句子和实体的表示，并且通过前馈神经网络对它们进行输入，从而获得用于分类的

最终表示：

$$\boldsymbol{h}_{\text{final}} = \mathbf{FFNN}([\boldsymbol{h}_{\text{sent}}; h_s; h_o]) \tag{7.60}$$

将最终得到的隐藏表示输入到一个线性层中，然后再经过一个 softmax 操作去获取关系的概率分布。最后，模型根据这个概率分布去推测出输入实体的最合适的关系。

7.3.2 远程监督关系抽取

有监督的关系抽取方法虽然准确率较高，模型结果更为可靠，但需要人工标注数据集，构造这样的数据集需耗费大量的人力和物力。由于关系种类相较于实体种类更加复杂多样，并且不断涌现，针对所有关系抽取都需要预先标注大量样本，制约了关系抽取的更广泛的应用。近年来，为了实现自动化关系抽取，研究人员们提出了**远程监督**（Distant Supervision）方法。远程监督假设知识库中两个实体存在某种关系，那么所有提及了这两个实体的句子都表达了这种关系。图7.13展示了通过远程监督方法自动标注的实例，在这个例子中，“Apple”和“Steve Jobs”是 Freebase 知识库中两个实体，其关系为“/business/company/founders”，那么在语料中所有包含这两个实体的句子都被标注为这种关系并作为训练样本用于训练。

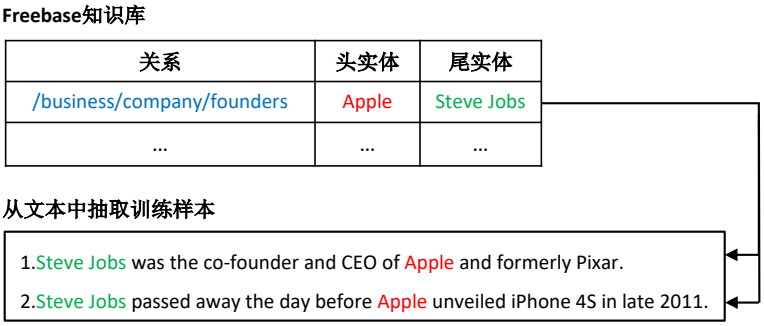


图 7.13　远程监督方法生成的训练样本示例

虽然利用远程监督方法可以迅速构建大量训练样本，但是远程监督方法构建标签的假设过强，导致大量的错误标注，从而严重影响模型的性能。图7.13中给出的例子中也可以看到，虽然句子 2 中包含了“Apple”和“Steve Jobs”两个词语，但是该句并没有表达“/business/company/founders”关系。因此，大量的远程监督方法关注于如何缓解训练集中的噪音信号来获得高性能的抽取器。本节将分别介绍远程监督关系抽取的两个经典方法：多示例学习方法和基于注意力的方法。

1. 基于多示例学习的远程监督关系抽取

多示例学习是指对具有某种特征的数据样本集合进行标注，这样的样本集合称为包（Bag），模型在包级别上进行训练与推断。其形式化定义为给定示例集合 $X = \{x_1, x_2, ..., x_n\}$，根据某种映

射 f_1，将示例集合映射到包 $B=\{B_1,B_2,...,B_M\}$，即将提及相同实体对的示例映射到同一个包中，然后经过机器学习或深度学习模型 f_2，将包映射到标签空间 $L=\{L_1,L_2,...,L_T\}$。多示例学习可以用于缓解远程监督关系抽取中的错误标注问题。

PCNN（Piecewise Convolutional Neural Networks）[18] 是一种应用了多示例学习来处理远程监督关系抽取的模型，其神经网络结构如图7.14所示，主要分为四部分：向量表示层、卷积层、分段最大池化层和输出层。

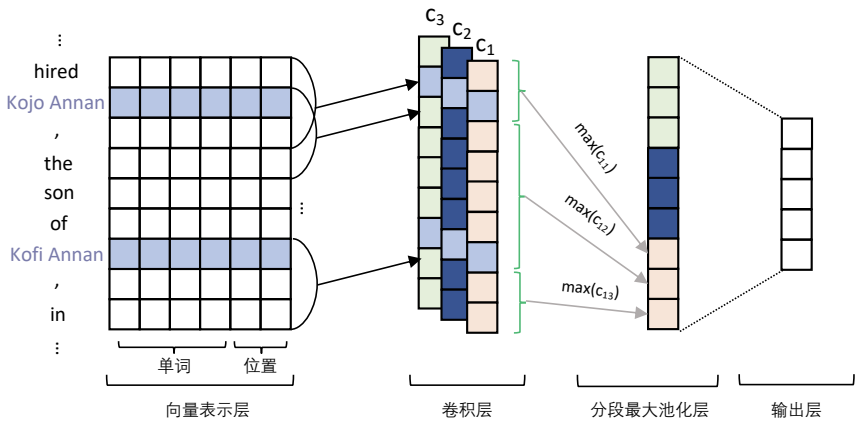


图 7.14　PCNN 模型结构[18]

向量表示层用于将词转化为低维的向量表示，从而输入到神经网络中。向量表示分为两部分，分别是词向量和位置向量，经过预训练的词向量可以捕获词的语法和语义信息，从而更好地适用于下游任务，位置向量用于编码头尾实体的位置信息。词向量和位置向量拼接后作为向量表示输入到模型中。

卷积层用于对句子的局部特征进行建模，卷积操作的计算如下所示：

$$c_{ij}=\boldsymbol{w}_i\boldsymbol{q}_{j-w+1:j},1\leqslant i\leqslant n \tag{7.61}$$

其中 $\boldsymbol{w}_i$ 为第 i 个卷积核，$\boldsymbol{q}_j$ 为输入句子中第 j 个词对应的向量表示，n 为卷积核数量。

分段最大池化层根据卷积操作得到的局部特征提取全局特征。不同于单一最大池化方法，分段最大池化方法首先以头尾实体为界将卷积得到的特征图划分为三部分，如图7.14所示，每个卷积核输出的特征图 $\boldsymbol{c}_i$ 被实体“Kojo Annan”和“Kofi Annan”划分为三部分 $\{\boldsymbol{c}_{i1},\boldsymbol{c}_{i2},\boldsymbol{c}_{i3}\}$，然后对每一部分分别进行最大池化操作，各特征图进行分段最大池化后拼接，经过非线性函数后得到全局

的特征表示。分段最大池化的具体计算公式如下：

$$p_{ij} = \max(c_{ij}), 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant 3 \tag{7.62}$$

$$p_i = \{p_{i1}, p_{i2}, p_{i3}\} \tag{7.63}$$

$$g = \tanh(p_{1:n}) \tag{7.64}$$

输出层根据全局特征使用 Softmax 激活函数计算在各关系类别上的概率分布，从而对关系类别作出预测。

为了解决远程监督带来的错误标注问题，PCNN 使用了多示例学习方法，所使用的包级别损失函数定义如下：

$$\mathcal{J}(\theta) = -\sum_{i=1}^{T} \log p(y_i|m_i^j;\theta) \tag{7.65}$$

其中，j 有如下约束：

$$j^* = \arg\max_j p(y_i|m_i^j;\theta), 1 \leqslant j \leqslant q_i \tag{7.66}$$

即 m_i^j 为每个包 $M_i = \{m_i^1, m_i^2, ..., m_i^{q_i}\}$ 中在正确标签上输出概率最高的示例。

多示例学习的完整训练过程为：

(1) 初始化网络参数 θ，将所有包划分为若干个大小为 b_s 的小批次（mini-batch）。

(2) 随机选择一个批次，根据公式7.66选出包中第 j 个示例 m_i^j $(1 \leqslant i \leqslant b_s)$。

(3) 基于示例 m_i^j 的梯度对网络参数 θ 进行更新。

(4) 重复以上两步直至收敛。

由此可见，传统的反向传播算法对所有的训练样本计算梯度进而更新网络参数，而多示例学习选择包中最符合包标签的样本对参数进行优化，在一定程度上过滤了远程监督中的噪音标签数据。

2. 基于注意力的关系抽取

多实例学习在远程监督的关系抽取方面有了很大的改进，但模型只取置信度最高的一个句子进行训练的方法会造成大量丰富信息的丢失。同时模型将包含相同实体的语句作为一个包，这种基于包级别的训练和推断仍然存在引入过多噪声的问题。

为了解决上述问题，Lin 等人[19] 提出了基于句子级别的注意力机制模型，将注意力机制应用到远程监督关系抽取中。该模型通过使用卷积神经网抽取句子的语义特征，然后在多实例上构建句子级别的注意力机制，从而动态减少噪声实例的权重并全面的获取实例的信息。给出一组句子 $\{x_1, x_2, ..., x_n\}$ 和两个相对应的实体，模型评估每个关系 r 的可能性。模型主要包含两大部分：句子编码器和实例选择注意力。其结构神经网络结构如图7.15所示。

句子编码器包括词嵌入，卷积层，最大池化层和非线性层。给定一个句子 x 和两个目标实体，

卷积神经网络用来构建句子的分布表示。首先, CNN 的输入是句子 x 的原始单词, 将每个输入词通过词嵌入矩阵转换成不同的低维向量。为了指定每个实体对的位置，还对句子中的所有单词使用位置嵌入。使用卷积层来合并处理这些特征, 处理方法和 PCNN 中处理方法相同, 如公式7.61所示。

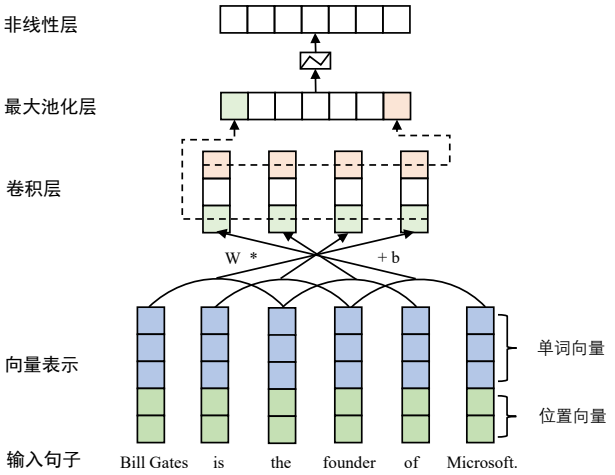


图 7.15 CNN/PCNN 模型句子编码器结构[19]

实例选择注意力使用句子级别注意力去选择表达了相应关系的句子。假设对于实体对（Head, Tail），存在于包含 n 个句子的集合 $B = \{x_1, x_2, \ldots, x_n\}$ 中。为了充分利用所有句子的信息，同时缓解噪音句子带来的负面影响，模型给每个句子赋予一个可学习的注意力权重，最终集合 B 表示为所有句子向量 x_i 的加权和：

$$\boldsymbol{b} = \sum_i \alpha_i \boldsymbol{x}_i \tag{7.67}$$

其中 α_i 是每个句子向量 $\boldsymbol{x}_i$ 的权重，这里使用选择性注意力来计算，因为如果在训练和测试过程中将每个句子等分，标注错误的句子会带来大量的噪音。因此，使用选择性注意力来弱化噪声语句。α_i 被进一步定义为：

$$\alpha_i = \frac{\exp(e_i)}{\sum_k \exp(e_k)} \tag{7.68}$$

其中 e_i 被称为基于查询的函数，其对输入句子 X_i 和预测关系 r 匹配的程度进行评分，这里采用双线性形式作为查询函数：

$$e_i = \boldsymbol{x}_i \boldsymbol{A} \boldsymbol{r} \tag{7.69}$$

其中 $\boldsymbol{A}$ 是加权对角矩阵，$\boldsymbol{r}$ 是与表示关系 r 相关联的查询向量。

最后，通过 Softmax 层定义条件概率 $P(r|B,\theta)$ 如下：

$$P(r|B,\theta) = \frac{\exp(o_r)}{\sum_{k=1}^{n_r} \exp(o_k)} \tag{7.70}$$

其中 n_r 是关系的总数，$\boldsymbol{o}$ 是神经网络的最终输出，它对应于与所有关系类型相关联的分数，其定义如下：

$$\boldsymbol{o} = \boldsymbol{M}\boldsymbol{b} + \boldsymbol{d} \tag{7.71}$$

这里 $\boldsymbol{d} \in \mathbb{R}^{n_r}$ 是偏差向量，$\boldsymbol{M}$ 是关系的表示矩阵。

基于注意力的方法在多示例学习的基础上，摒弃了只采用最高置信度的句子作为训练语句的方法，而是综合考虑所有包含同一实体对的句子，对它们与预测关系的相关性给予不同的权重，进而得到更为全面、有用的信息。在充分利用包含每对实体的所有信息句的同时，通过选择注意力机制缓解了远程监督中带来的错误标签的影响。

7.3.3 开放关系抽取

传统的关系抽取算法，不论是有监督和远程监督关系抽取方法，目标都是针对预先定义的关系类型在限定语料中判定实体之间是否是否存在预先定义的关系。因此，传统的关系抽取算法能够处理的关系数量有限，并且在处理不同切换领域时需要用户进行关系类别定义、数据标注以及模型训练等一系列工作。在处理海量互联网和社会媒体数据时，传统关系抽取算法受到上述问题的制约，很难适应快速发展且不断演进的需求。**开放关系抽取**（Open Relation Extraction，ORE）任务目标是在不需要预先关系定义的情况下，从非结构化文本中提取关系元组，并且不受语料库领域的限制。实体关系仍然采用三元组形式表示：（Head，Relation，Tail）。

例如：根据句子“卡塔尔发布本届世界杯吉祥物，正式取名叫做 La’eeb。”

可抽取 < 世界杯，在，卡塔尔 >、< 世界杯吉祥物，是，La’eeb> 等关系

需要特别注意的是，与预定义关系抽取不同，开放关系抽取所得到关系类型描述并不是预先定义的，而是根据所给定的文本截取或生成的。因此，具有相同语义的关系类型可能存在多个描述形式。

本节将分别介绍基于抽取和基于聚类两种开放关系抽取方法。

1. 基于人工特征的开放关系抽取

2007 年 Banko 等人[20] 提出了开放信息抽取（Open IE，OIE）的概念，并设计了 TextRunner 系统，试图打破传统的封闭式信息抽取系统，使用一种无监督的方式，提取出类型更加多样的关系元组。TextRunner 采用自监督学习方法，通过自监督的学习器、信息抽取器和基于冗余信息的评估器等三个主要部分完成开放信息抽取：

(1) **自监督学习器**（Self-Supervised Learner）：利用从整个语料集合中采样得到的小规模样本，根据自监督学习方法输出分类器，目标是对所有可能的抽取候选分类为“可信”（Trustworthy）或“不可信”（Not Trustworthy）。

(2) **信息抽取器**（Single-Pass Extractor）：对整个语料集合中所有数据进行遍历并抽取出所有的关系三元组，并利用自监督学习器构建的候选分类器进行判别，保留分类为“可信”的三元组。

(3) **基于冗余信息的评估器**（Redundancy-Based Assessor）：利用文献 [21] 所提出的模型利用文本中的冗余信息对三元组进行评价。

自监督学习器使用依存句法分析器解析句子的句法结构，抽取名词性短语组成三元组 <Head, Relation，Tail>，并利用人工定义的规则将所抽取的三元组划分为正样本或负样本。如果三元组满足以下三个条件则被归类为正样本，否则被归类为负样本：

- Head 和 Tail 之间不超过一定长度；
- Head 和 Tail 之间的路径不穿过句式边界（例如关系从句）；
- Head 和 Tail 不全是代词。

在此基础上，利用人工定义的特征类型，针对所有三元组构造特征向量，并利用朴素贝叶斯分类器用来构建分类器。

信息抽取器对整个语料库进行一次遍历，对所有句子进行词性标注，并使用轻量级的名词短语分块器来识别名词短语，进而识别出实体。关系则是通过分析实体之间的文本，通过人工定义的一系列启发式规则消除非必要的短语构成。

例如：Scientists from many universities are studying. 简化为 Scientists are studying.

definitely developed 简化为 developed

提取器从每个句子中生成一个或多个候选元组，并将每个候选元组利用自监督学习器所输出的分类器进行分类，仅保留那些被标记为可信的三元组。

在对整个语料库完成提取操作后，TextRunner 将实体和规范化关系都相同的三元组进行合并，并统计每个三元组在不同句子中出现的次数。利用上述信息，基于文献 [21] 所提出信息冗余的概念，对被抽取多次的三元组分配一个更高的置信度，说明被抽取多次的三元组更有可能表示为一种关系。最终根据置信度和预先设定的阈值输出所发现的实体和关系。

TextRunner 开放关系抽取系统，虽然减少了人工标注训练数据的开销，并且可以在多个不同领域的语料上使用，但是仍然存在不足：

- 不连贯的抽取（Incoherent Extractions）：抽取的关系词语不连贯，并且没有可解释性的意义。这样的关系类别占据了 TextRunner 大约 13% 的输出。

  例如：The guide contains dead links and omits sites.

  TextRunner 系统抽取的关系类型：“contains omits”；
- 无意义的关系（Uninformative Relations）：抽取忽略了关键性的信息，没有处理好动词和名词组成的多词谓语，并且名词携带了谓词的语义信息。

  例如：Hamas claimed responsibility for the Gaza attack.

  TextRunner 系统抽取结果：（Hamas, claimed, responsibility）

  正确结果：（Hamas, claimed responsibility for, the Gaza attack）

针对以上的问题，2010 年 Fader 等人提出了 Reverb 算法[22]，利用句法约束和词汇约束对抽取三元组进行限制。句法约束目的是为了排除不连贯的抽取的问题，同时通过动词结构捕捉关系短语。具体句法约束如下所示：

V | V P | V W* P

V = verb particle? adv?

W = (noun | adj | adv | pron | det)

P = (prep | particle | inf. marker)

上述约束，给出了关系短语需要满足的词性标签，将关系短语限制为简单的动词短语、动词短语紧跟介词或小品词、动词短语接名词短语并以助词结尾这三种形式。如果在一个句子中有多个匹配则选择最长的匹配。如果该模式匹配多个相邻的序列，则将它们合并为一个单一的关系短语。这种细化匹配使得模型能够处理包含多个动词的关系短语，并且关系短语必须是句子中一个连续的单词片段。

句法约束大大减少了无意义的关系的提取，但在一些特定的情况下，非常复杂的关系短语也能够满足匹配，例如："The Obama administration is offering only modest greenhouse gas reduction targets at the conference." 中 "is (verb, V) offering (verb, V) only (Adv, W) modest (Adj, W) greenhouse (Noun, W) gas (Noun, W) reduction (Noun, W) targets (Noun, W) at (Prep, P)" 符合句法约束，但并不具有实际意义。为了克服这一缺陷，Reverb 算法还引入了词汇约束，用于避免抽取过度冗长的关系短语，保留有效的关系短语。该约束基于的假设是：一个有效的关系短语应该在一个大型语料库中出现过许多次，有许多不同的实体对。上个例子中的冗长短语几乎不可能存在于多个实体对，所以不能代表一个真实的关系。

2. 基于聚类的开放关系抽取

在开放环境中，我们无法预先定义好所有可能的关系类别标签。但是即便不知道某个关系属于哪一类，仍有可能判断一系列的关系是否是同一类。因此在开放环境的背景下，研究人员们提出了基于聚类的关系抽取。此类模型的核心目标是构建在很小甚至无需人类标注的前提下学习到更好的关系语义表示模型，从而在新的关系类别中识别关系三元组。基于聚类的开放关系抽取基本的任务实现流程如图7.16所示。主要包含三个部分：实体对编码、编码非线性映射以及关系聚类。

实体对编码目标是需要从文本中获得关系三元组的表示，这一步骤通常使用 BERT、RoBERTa 等经过较好预训练后的文本编码器来实现。对于一个输入文本，通常编码器将输出多个向量表示，对于 BERT 模型而言，通常取第一个 Token 的标记 '[CLS]' 的输出向量，作为整个句子的语义表示。但是针对开放关系抽取需要特别注意的是：一段文本中可能不止包括一对三元组的关系信息，同时还夹杂着许多对于关系抽取任务而言的无关信息，例如："ChatGPT 是由 OpenAI 开发的一个人工智能聊天机器人程序，于 2022 年 11 月推出，使用基于 GPT-3.5 架构的大型语言模型并通过强化学习进行训练。"。对于 <'ChatGPT'，'开发'，'OpenAI'> 的关系三元组而言，其余文本信息都是与该关系无关的冗余信息，因而对于关系抽取任务而言，[CLS] 位置所对应的向量并不是表

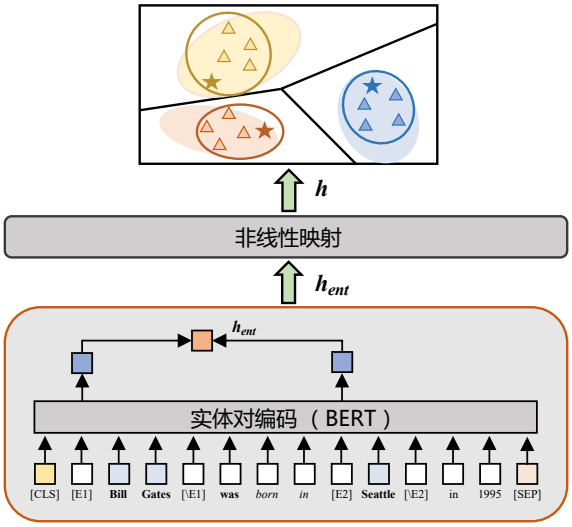


图 7.16　基于聚类的开放关系抽取基本流程

示关系三元组的最好选择。针对此问题可以采用实体标记（Entity Mention）的方法，在数据预处理阶段，将文本中需要抽取关系的实体对用特定的符号进行标记，将标记位置的向量输出作为关系三元组的向量表示。如图7.16所示，Bill Gates 和 Seattle 作为实体对词，前后都加入了特定符号[E]。利用这种方法可以使得编码器在输出句子的关系表示时能聚焦于关键信息，从而更好的概括句子级别的语义。具体过程可以形式化表示为：

$$X = [x_1, \ldots, [E1_{\text{start}}], x_i, \ldots, x_{j-1}, [E1_{\text{end}}] \ \ldots, [E2_{\text{start}}], x_k, \ldots, x_{l-1}, [E2_{\text{end}}], \ldots, x_T] \tag{7.72}$$

其中 x_i 到 $x_j - 1$ 以及 x_k 到 $x_l - 1$ 分别是头尾实体，该句子最终的关系向量表示为：

$$\boldsymbol{h} = \left[\boldsymbol{h}_{[E1_{\text{start}}]}, \boldsymbol{h}_{[E2_{\text{start}}]}\right] \tag{7.73}$$

在将文本编码成向量的关系表示后，通常需要将所有关系向量都投影到相同维度的向量空间中以便聚类。在此步骤可以针对编码器的输出结果经过一个非线性的映射层（如 MLP 等方法）来实现从高维到低维的投影。在此步骤中也可以引入 dropout 等方法，使得模型的聚类效果更鲁棒。非线性映射层具体可以采用如下步骤实现：

$$\widetilde{\boldsymbol{h}} = \text{Dropout}(\boldsymbol{h}) \tag{7.74}$$

$$\boldsymbol{z} = \text{g}\left(\boldsymbol{W}_\phi \widetilde{\boldsymbol{h}} + \boldsymbol{b}_\phi\right) \tag{7.75}$$

通过实体对编码和非线性映射两个步骤，所有输入句子将离散的嵌入到同一个低维的特征空间内。在此基础上，可以通过 K-Means 等聚类算法来为空间内的关系点聚簇，并为每个点分类一个伪标签：

$$\hat{y}^u = \text{K-Means}\left(\boldsymbol{h}^{u\prime}\right) \tag{7.76}$$

最后针对不同簇内的样本内容以及对应的伪标签，人工判断此簇所代表的关系。这种方法对于聚类结果的效果要求较高。针对该问题，SelfORE[23] 采取自监督的方法，根据样本的嵌入点与聚类质心的距离来为每个样本分配一个置信度，高置信度的数据可以将其伪标签作为监督数据来训练模型。基于 t 分布的置信度算法如下：

$$q_{nk} = \frac{\left(1 + \left\|\boldsymbol{z}_n - \mu_k\right\|^2 / \alpha\right)^{-\frac{\alpha+1}{2}}}{\sum_{k'}\left(1 + \left\|\boldsymbol{z}_n - \mu_{k'}\right\|^2 / \alpha\right)^{-\frac{\alpha+1}{2}}} \tag{7.77}$$

其中 α 是超参数，用于衡量 t 分布的自由度。μ_k 是类的质心，z_n 是样本的嵌入向量，$q_n k$ 便是置信度。

开放领域的实体抽取中如何能够对于句子关系语义进行很好的表示是这类方法研究的核心。但是关系语义非常复杂，完全依赖无监督表示构建方法通常不能够精准地捕捉关系间的语义相似性。为解决上述问题，RoCORE[24] 通过利用相对容易获得的预定义关系类别标注数据，进一步增强关系表示学习，并利用多目标联合训练有效地减少预定义类别的偏置并优化实体对表示。

7.3.4 关系抽取评价方法

针对预先定义的关系类型，通常采用的评价指标与命名实体识别一样，包括精度（Precision，P）、召回率（Recall，R）、F 值（F-Measure）等指标对不同类型关系进行评价，也可以利用准确率（Accuracy）、微平均 F1（Micro-F1）、微平均精度（Micro-P）、微平均召回（Micro-R）等对整体抽取效果进行评价。

而对于开放关系抽取，常用的评价指标有 V-measure、调整兰德系数（ARI）等。V-measure 是聚类任务中常用的评价指标，其基于两个类别之间的条件熵计算之间的同质性和完整性。ARI 是聚类任务中描述簇类相似度的指标。具体计算公式如下：

(1) **同质性**（Homogeneity）度量：

$$
\begin{aligned}
h &= 1 - \frac{H(C|K)}{H(C)} \\
H(C|K) &= -\sum_{c=1}^{C}\sum_{k=1}^{K}\frac{n_{c,k}}{n}\log(\frac{n_{c,k}}{n}) \\
H(C) &= -\sum_{c=1}^{C}\frac{n_c}{n}\log(\frac{n_c}{n})
\end{aligned}
\tag{7.78}
$$

其中 $H(C|K)$ 是给定划分条件下类别划分的条件熵，$H(C)$ 为类别划分熵，n 表示全部实例数，n_c 表示类别 c 下的实例数，n_k 表示在簇 k 下的实例数，$n_{c,k}$ 表示类别 c 中被划分到簇 k 下的实例数。

(2) **完整性**（Completeness）度量：

$$
c = 1 - \frac{H(K|C)}{H(C)} \tag{7.79}
$$

(3) **V-measure** 是同质性和完整性的调和平均值：

$$
v = \frac{2 \times h \times c}{h + c} \tag{7.80}
$$

7.3.5 关系抽取语料库

自 1998 年 MUC-7 会议上首次正式提出关系抽取任务以来，无论 1999 年到 2008 年举行了 9 届的 ACE 会议，还是自 1998 年一致举办至今的 SemEval 会议都对关系抽取任务给于了很大的关注，发表了一些列限定域和开放域的关系抽取评测集合。表7.3展示了常用的关系抽取语料库，主要包括: SemEval-2010 Task 8、NYT-10、TACRED 和 FewRel。

数据集	关系类别数目	关系实例数目	应用域
SemEval-2010 Task 8	9	10,717	限定域
TACRED	42	21,784	限定域、开放域
FewRel	100	70,000	限定域、开放域
NYT-10	57	143,391	限定域、开放域

表 7.3　常见关系抽取语料库汇总

1. SemEval-2010 Task 8 关系抽取语料集

SemEval-2010 Task 8 数据集[25] 主要用于限定领域关系抽取任务评价，自于 2010 年的国际语义评测大会中（SemEval）Task 8：“Multi-Way Classification of Semantic Relations Between Pairs of

Nominals”。该数据集包含 10717 个样本，其中 8000 个用于训练，2717 个用于测试，标签集包含 18 种有序关系类型和 1 种未知关系类型。

2. TACRED 关系抽取语料集

TACRED（TAC Relation Extraction Dataset）数据集[26] 是一个拥有 106264 条实例的大规模关系抽取数据集，这些数据来自于每年的 TAC KBP（TAC Knowledge Base Population）比赛使用的语料库中的新闻专线和网络文本。该数据集 41 种已知关系类型和 1 种未知关系类型，是一个典型的长尾分布数据集，其中标签为未知关系类型的数据占据了 79.5%。

3. FewRel 关系抽取语料集

FewRel 数据集[27] 包含 100 个类别、70,000 个实例，是目前采用人工标注的关系抽取任务中关系类别和实例数目最大的语料库。由于该语料库关系数量的多，FewRel 语料集合也经常应用于小样本学习（Few-shot learning）和远程监督关系抽取任务中。

4. NYT-10 关系抽取语料集

NYT-10 数据集[28] 是在基于远程监督的关系抽取任务上最常用的数据集。其文本来源于纽约时报 New York Times，命名实体是通过 Stanford NER 工具并结合 Freebase 知识库进行标注的。命名实体对之间的关系是链接和参考外部的 Freebase 知识库中的关系，结合远监督方法所得到的。数据集中一共包含 52 个已知关系类型和 1 个未知关系类型。

7.4 事件抽取

事件抽取（Event Extraction）目标是从文本中发现特定类型事件，并抽取该事件所涉及的时间、地点、人物等元素。事件抽取任务可以为问答系统、文本摘要以及各类语言理解任务提供有效的结构化信息。根据美国国家标准技术研究所（NIST）组织的 ACE（Automatic Content Extraction）项目给出的定义[13]，事件由事件触发词（Trigger）以及事件论元（Argument，也称事件元素）组成。

例如：2022 年卡塔尔世界杯（FIFA World Cup Qatar 2022）是第二十二届国际足联世界杯足球赛，在当地时间 2022 年 11 月 20 日到 12 月 18 日间在卡塔尔国内 5 个城市的 8 座球场举行。

事件类型：体育赛事

触发词：举行

赛事名称：第二十二届国际足联世界杯足球赛

时间：2022 年 11 月 20 日到 12 月 18 日

地点：卡塔尔

根据事件信息是否预先定义，事件抽取可分为限定域事件抽取和开放域事件抽取两种类型，本节将针对这两种类型的事件抽进算法行介绍。

7.4.1 限定域事件抽取

限定域事件抽取会预先定义事件类型以及与之对应的事件论元，抽取算法的目标就是从包含事件的文本中识别特定类型的事件并提取相应的事件论元。**事件类型**是标识事件的类别，常用的 ACE2005 数据集包括 8 种事件类型，33 种子类型，如“会议”、“袭击”等。**事件触发词**是指最清楚和明显地表达事件发生的主要词，如“击打”、“结婚”等。**事件论元**是指事件中涉及的参与者，一般为实体、时间等。**论元角色**（也称元素角色）是指事件论元在事件中所扮演的角色。

例如：句子 “小明 2022 年在上海与小李举行婚礼” 中，由事件触发词 “举行婚礼” 可以得到事件类型为 “结婚” 事件，人物 “小明” 和 “小李”、时间 “2022 年”、地点 “上海” 都为事件论元，对应 “结婚” 事件模板中的论元角色分别为 “结婚的人”、“结婚时间”、“结婚地点”。

限定域事件抽取系统进行事件抽取，可以分解为事件类型识别、事件论元抽取等多个子任务进行，也可以采用联合抽取框架以减少错误传递。

1. 基于分类的事件抽取方法

David Ahn 在 2006 年提出了将事件抽取任务分解成一系列的分类子任务的方法[29]，并针对不同子任务提出了特征构建方法，在基于记忆的分类算法 TiMBL[30] 以及最大熵分类算法 MegaM[31] 上分别进行验证。该方法将事件抽取任务转换为以下四个子任务：

1. 触发词识别：从文本中识别触发词，并根据触发词确定事件提及（Event Mention）类型；
2. 论元识别：针对每个事件提及，从文本中识别事件提及的相关论元，包括实体、时间等；
3. 属性分配：确定每个事件提及的模态、极性、概括性和时态等属性；
4. 事件共指：确定从文本发现的事件提及为同一事件。

对于以上子任务，文献 [29] 所提出的算法采用了流水线处理框架，其中触发词识别独立于其他任务，论元识别和属性分配任务依赖于触发词识别的结果，在这三个子任务都处理完成后，进行事件共指判断。本节中将主要介绍事件抽取中最关键的触发词识别和论元识别两个子任务。

针对事件触发词识别，由于在事件描述中，触发词往往由明显的单个词语组成，在 ACE2005 数据集中，就有超过 95% 的触发词都为单个单词，所以将触发词识别转换为词分类任务。但是，由于文本中词语的数量非常多，逐个分类会非常影响触发词的提取效率，并且这种情况下正负例比例相差过于巨大。因此，在该方法中考虑触发词词性。由于触发词词性通常是由名词、动词、形容词组成，所以首先对文本进行词性标注得到词语的词性信息后，再对相应词性的词进行分类。触发词识别主要由两个阶段组成：(1) 采用二分类分类器，将经过词性标注筛选后的词依次输入在训练集中训好的二分类分类器中，判断该词是否为触发词；(2) 采用多分类模型判断候选触发词的类型。

在事件论元识别方面，事件论元识别可以简化为一个成对分类（Pair Classification）任务，将包含事件描述的句子与同句中的事件论元内容组成待分类对，再利用分类模型判断论元角色。根据预先给定的事件定义，特定事件类型中有固定的论元角色，比如在“攻击”事件中包含攻击者、攻击目标、攻击时间、攻击发生地等几种固定论元角色。由于在进行事件论元识别已经通过触发词

判别得到了事件类型，可以针对每个事件类型来训练不同的分类器来得到更好的效果。

文献 [29] 采用了基于特征的分类方法，针对上述分类任务，对输入内容利用人工设计的特征构建特向向量表示，并利用有监督分类算法进行建模。在本任务中，利用了单词特征、WordNet 特征、上下文特征、依存句法特征、实体相关特征，针对不同的任务设计了不同的特征抽取策略。具体的特征描述可以参考文献 [29]。

2. 基于循环神经网络的联合事件抽取方法

采用流水线框架，将事件抽取分解为多个子任务的模式容易造成错误传递的问题，同时传统机器学习方法还需要依赖于预先设计好的语言工具来提取句子中的词汇和上下文特征。这种对预处理工具的依赖使得以往的事件抽取模型缺乏主动捕捉这些隐藏信息的能力，从而限制了这些模型的通用性。因此，研究人员们进一步提出了基于神经网络的联合事件抽取方法。在联合事件抽取方法中，模型需要自主地标记事件触发词的位置，并对标记出的事件类别进行预测。这种改进有助于消除模型对预处理的过度依赖以及流水线架构的错误传递问题问题，并获取更加通用的事件抽取模型。下面我们通过一个实际的例子来了解联合抽取任务与之前任务的差异和必要性。

一般来说，可以将语句中的特征分为两类: 词汇特征和上下文特征。词汇特征包括词性、实体信息和形态学标记，如词形（Token）和词根（Lemma），主要应用于捕捉语义或是单词的背景知识。例如：考虑下述的三个句子中的句 1 和句 2，它拥有一个存在歧义的单词 beat：

句 1：Obama beats McCain.

句 2：Tyson beats his opponent.

句 3：In Baghdad, a cameraman died when an American tank fired on the Palestine Hotel.

在句 1 中，“beat”是一个类型为“选举事件”的触发词。然而，在句 2 中“beat”是一个类型为“攻击事件”的触发词。由于在现实环境中，单词“beat”作为攻击事件类型比作为选举事件类型更为常见，因此传统的方法可能会错误地将句 1 中的“beat”标记为攻击世间的触发词。但是，如果我们事先知道奥巴马和麦凯恩都是总统候选人，我们就有更充分的证据来认为“beat”是选举类型的触发因素。类似的知识也可以称为词级别的线索（Lexical-level Clues）。

除了词汇特征，一些方法还尝试通过上下文特征来了解事实是如何从更大的视角联系在一起的。上下文特征有助于模型对事件和论点进行更精确的识别。图7.17展示了句 3 的事件提及和语法解析结果。图中显示两个事件提及共享了三个论元：由“died”触发的“死亡事件”，以及由“fired”触发的“攻击事件”。图下方显示了对应的依赖结果。通过事件论元“cameraman”与触发词“died”之间的“nsubj”依赖关系，可以归纳出在死亡事件中“cameraman”的受害者角色，这些信息就是一种句子级线索。值得注意的是，事件论元“cameraman”和它的触发词“fired”存在于不同的子句中，并且在它们之间不存在直接的依赖路径。因此，传统方法很难通过上述的依赖特性在它们之间找到正确的依赖关系。

通过上述分析，我们可以看到传统事件抽取方法有两个重要缺陷：

(1) 错误传播：传统的方法依赖于预处理过程中对事件触发词的标注结果。如果对事件触发词的

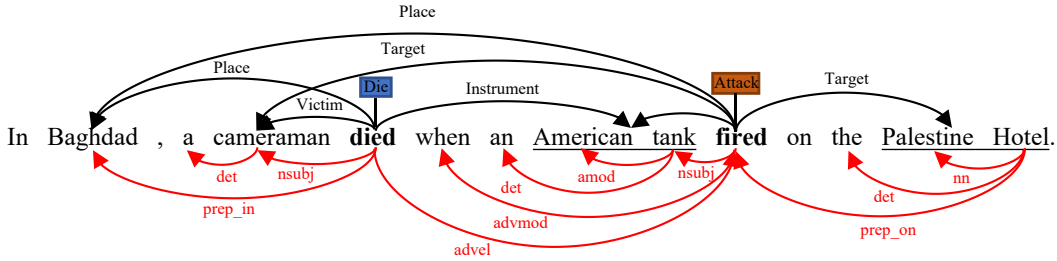


图 7.17　句 3 的事件提及和语法解析结果

标注发生了错误或者遗漏，模型就无法主动修正标注带来的噪声，只能在错误的数据上进行训练和预测。这些错误标注会严重影响模型在任务上的表现。

(2) 数据稀疏：传统的事件抽取模型仅关注被标注的事件实体，无法主动提取和处理句子中未获得标注的实体成分。然而，就像句 3 所展示的那样，在事件提取任务中一个句子可能包含两个或多个事件，这些事件以不同的实体共享相同的事件论元。在经典的 ACE2005 数据集中，这样的多事件句子占数据集中所有数据的 27.3%。传统训练方式只关注了这些句子中的部分事件信息，因此也降低了模型的训练效果。

这两个问题使得传统事件抽取方法对于标注中未见的实体和特征的泛化能力很差，通用性亟待提高。导致上述问题的主要因素是模型过度依赖于预处理工具提供的事件触发词信息，为了解决这一缺陷，研究人员们提出了联合事件抽取方法来统一地处理事件触发词抽取和事件分类步骤。

文献 [32] 中提出了通过卷积深度神经网络（CDNN）来提取词汇和上下文层次的特征。输入句子中的词编码，位置编码和事件类型后，模型能够自动提取词汇级和句子级特征来标记事件触发词的位置，在一定程度上解决了数据稀疏的问题。在标记了事件触发词后，分类器将事件触发词分类到具体的事件类别中。总的来说，模型采用两步结构，通过先识别再分类的方法来联合处理事件标记和事件分类。但这种流水线架构还是不能很好地缓解错误传播的问题，一旦对事件触发词的识别发生了错误或遗漏，就会大大影响后续的分类正确率。此外，这个方法没有尝试利用事件触发词和事件论元的依赖信息，也没有考虑到不同事件论元之间的相关性。

为了解决上述问题，Nguyen 等人在 2016 年提出 JRNN 方法[33]，使用循环神经网络（RNN）来获取句子中不同事件触发词和事件论元间的长距离依赖关系。JRNN 在保持对预处理的低依赖性的同时，提升了模型对不同事件之间重叠参数的识别和利用。模型使用两个 RNN 分别正/反向学习句子的表示，从而完成自动构建特征的工作；另一方面，JRNN 使用了一个记忆向量与两个记忆矩阵来存储事件触发词与事件论元之间的依赖关系，引入了对离散的事件论元特征的长距离建模，有效解决了 CDNN[32] 方法存在的问题。模型的结构如图7.18所示，主要分为编码层和预测层两个模块。

编码层由句子编码层和基于 RNN 的特征编码层两部分构成，以图中的句子“A man died when

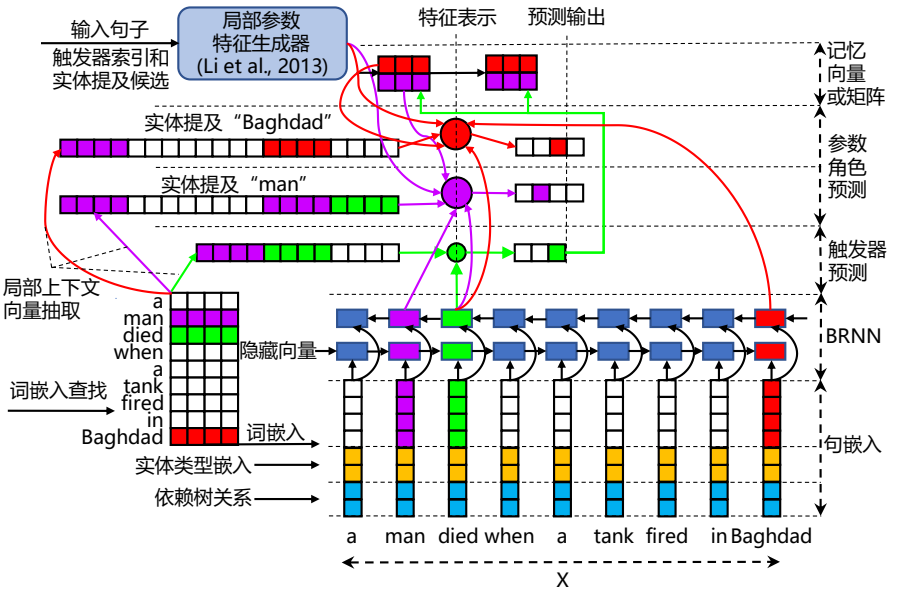


图 7.18 基于 RNN 的联合事件抽取模型[33]

a tank fired in Baghdad"为例，在句子编码层中，输入文本被转化成由三个向量的拼接而成的编码结果：

(1) 词编码（Word Embedding）：使用预训练的词嵌入表来获取每个词的向量表示

(2) 实体类型编码（Entity Type Embedding）：通过查找预训练的实体类型嵌入表，使用 BIO 注释模式来提供当前词的实体信息

(3) 依存关系编码（Dependency Relation Embedding）：使用训练得到的依存句法树获取某个词相对于其他词的依赖关系特征

在此基础上，使用双向 RNN 处理上述三个编码结果，首先词编码、实体编码和依赖关系编码拼接得到句子表示，再输出每个词对应的隐向量（hidden vectors）作为词信息和上下文信息的特征表示。形式化表示如下：

输入 $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)$，编码层包括两个方向相反的 RNN 网络层 $\overrightarrow{RNN}$ 和 $\overleftarrow{RNN}$ 后续的分类层的输入数据为编码后 X 的隐藏表示 $(\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_n)$，其中 $\boldsymbol{h}_i = [\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_i^{'}]$ 满足

$$\begin{aligned}(\boldsymbol{\alpha}, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_n) &= \overrightarrow{RNN}(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n) \\ (\boldsymbol{\alpha}_1^{'}, \alpha_2^{'}, ..., \boldsymbol{\alpha}_n^{'}) &= \overleftarrow{RNN}(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)\end{aligned} \tag{7.81}$$

通过上述的编码过程，输出向量被视作综合了词汇和上下文特征的句子表示。该表示随后被送入

预测层来获得对触发词语、论元角色和事件分类的预测结果。

为了在模型的预测层中联合预测触发词和论元角色，JRNN 利用额外的记忆向量来编码触发词标签和论元角色之间的依赖关系：使用二元记忆向量 $\boldsymbol{G}_i^{trg}$ 记录触发词之间的关联关系；使用二元记忆矩阵 $\boldsymbol{G}_i^{arg}$ 来记录论元间的关联关系；使用记忆矩阵 $\boldsymbol{G}_i^{arg/trg}$ 来记录触发词和论元之间的关联关系。矩阵中的每个论元表示行指标指代的实体与列指标指代的实体的相关程度，这些记忆向量和矩阵被统一地初始化为 0，并在训练过程中被不断更新。

根据事件抽取的任务需求，预测层共分为三个部分，分别为触发词预测层、论元预测层以及记忆向量和矩阵的存储区。下面将分别介绍触发词和论元的具体预测过程。

触发词预测(Trigger Prediction)过程在图中使用绿色的箭头表示。以识别句子中触发词“died”的过程为例，在编码层中，模型最终使用三个向量的拼接来作为当前单词的编码表示 $\boldsymbol{R}_i^{trg}$：

$$\boldsymbol{R}_i^{trg} = \left[\boldsymbol{h}_i, \boldsymbol{L}_i^{trg}, \boldsymbol{G}_{i-1}^{trg}\right]$$

其中 $\boldsymbol{h}_i$ 是上一步的隐藏向量（hidden vector），$\boldsymbol{L}_i^{trg}$ 是 w_i 的局部上下文，窗口大小为 d，$\boldsymbol{G}_{i-1}^{trg}$ 是上一步的记忆向量（memory vector）。

触发词预测层接收上述的编码结果，并将其作为输入来获取词“died”的置信度 $P_{i;t}^{trg}$。

$$P_{i;t}^{trg} = P_i^{trg}(l = t) = F_t^{trg}\left(R_i^{trg}\right)$$

同时，对于高置信度的词，模型将词的预测结果输入记忆矩阵（memory matrices）来保存。这些保存的表示将在之后的论元预测中被提取以作为上下文信息表示。

论元预测过程在图中使用红色和紫色的箭头表示。模型对每个实体 e_j 预测标签 a_{ij}，即在 w_i 的触发词类型为 t_i 的情况下，实体 e_j 对于该触发词的论元角色 a_{ij}。我们以实体“man”为例来解释模型的预测过程。模型使用实体对 $[e_j, w_i]$ 的编码作为输入来研究实体对于特定触发词的影响。根据图中紫色箭头的来源，具体地说，论元预测器的输入 R_{ij}^{arg} 由四个不同的分量构成，分别为：

(1) 触发词 w_i 的隐向量 $\boldsymbol{h}_i$ 和实体 e_j 的隐向量 $\boldsymbol{h}_{i_j}$
(2) 触发词 w_i 和实体 e_j 的上下文信息向量 $\boldsymbol{L}_{ij}^{arg}$
(3) 在之前学习过程中被记录，表示触发词和实体间相互关系的二元特征向量 $\boldsymbol{V}_{ij}$ 的隐向量 $\boldsymbol{B}_{ij}$
(4) 实体 e_j 的记忆向量的输出 $\boldsymbol{G}_{i-1}^{arg}[j]$ 和 $\boldsymbol{G}_{i-1}^{arg/trg}[j]$

预测器最终获取的输入向量 $\boldsymbol{R}_{ij}^{arg}$ 可以表示为：

$$\boldsymbol{R}_{ij}^{arg} = \left[\boldsymbol{h}_i, \boldsymbol{h}_{i_j}, \boldsymbol{L}_{ij}^{arg}, \boldsymbol{B}_{ij}, \boldsymbol{G}_{i-1}^{arg}[j], \boldsymbol{G}_{i-1}^{arg/trg}[j]\right]$$

对于预测器输出的结果向量，模型选取其中置信度最高的实体作为最终预测的论元实体。

综上，JRNN 模型结构的总体优化目标可以表示为：

$$
\begin{aligned}
C(T, A, X, E) &= -\log P(T, A \mid X, E) \\
&= -\log P(T \mid X, E) - \log P(A \mid T, X, E) \\
&= -\sum_{i=1}^{n} \log P_{i;t_i^*}^{\text{trg}} - \sum_{i=1}^{n} I\left(t_i \neq \text{ other }\right) \sum_{j=1}^{k} \log P_{ij;a_{ij}^*}^{arg}
\end{aligned}
\tag{7.82}
$$

这意味着模型的目标是尽可能准确地提取句子中的触发词和论元实体，并将每个触发词分配到该事件对应的两个正确论元上。

JRNN[33] 在多事件任务上表现出明显的效果，但在论元标签预测的效果上略不如 DMCNN[32] 结构的网络。此外，对于论元在不同子句之间的相关性的预测上，JRNN 相对之前的模型有所改善，但仍然存在识别和预测困难的现象，并没有完全解决事件抽取任务所面对的问题。在后续的研究中，我们仍然需要寻找更好的方法和思路。

7.4.2 开放域事件抽取

开放域事件抽取（Open Domain Event Extraction）其目标是在没有任何预定义域假设的情况下，从非结构化文本中挖掘提取有意义的事件信息。与限定域事件抽取任务不同，在没有预先定义的事件类型以及对应的事件论元情况下，早期开放域事件抽取目标不是精确地提取事件要素，而是聚类、语义分割等方法，对文本内容进行分析基础上检测并跟踪事件。近年来，也有一些工作试图给出更细粒度信息，针对给定的一系列文本内容，输出事件集合，以及每个事件的触发词和对应的事件论元列表。在本节中，我们将介绍基于聚类的开放域事件抽取方法，以及基于神经隐变量的细粒度开放域事件抽取方法。

1. 基于聚类的开放域事件抽取方法

基于聚类的开放域事件抽取目标是从无结构文本中抽取若干主题的相关内容组成一系列事件，可以分为两个主要类型：回顾事件抽取（Retrospective Event Extraction）和在线事件抽取（Online Event Extraction）。**回顾事件抽取**是将语料库中的文本内容进行分组，每一组文本被视为一个事件。**在线事件抽取**是在回顾事件抽取的基础之上，对当前时刻给定的文本进行实时处理，判断当前文本是已有事件还是新事件。回顾抽取和在线抽取分别采用离线和在线的聚类算法，本节中将介绍开放域事件抽取所采用两种聚类方法：

- GAC（**G**roup-**A**verage **C**lustering）多层次的聚类算法，用于回顾事件抽取。
- INCR（**IN**cremental **C**luste**R**ing）增量聚类算法，适用于回顾抽取和在线抽取。

文献 [34] 提出了一种事件检测和跟踪算法。使用基于词袋模型的传统向量空间模型对文档进行表示。每一篇文档（新闻报道）是通过一个带权重项的向量来表示，在聚类算法中是将所有文档的规范化向量表示用于聚类。文档向量是使用频率（TF）和逆文档频率（IDF）进行统计加权，并进行适当的标准化，同时只保留每个向量的前 k 项。

GAC 聚类算法采用自底向上的贪心并结合分而治之的策略，其目标是最大化聚类结果中每对文档之间的平均相似度。GAC 算法的基本流程是：(1) 将文档集合中的每个文档都作为一个单独的事件类；(2) 将当前事件类簇集合中的事件类按顺序连续并且不重叠地划分到 m 个组中；(3) 每个组内部进行局部聚类，重复地合并组中的 2 个最相似的事件类，直到桶中类数量减少的比例达到预设的阈值 p，或者任意 2 个类之间的相似度值均低于预设阈值 s 为止；(4) 将组之间的边界去除，并重复步骤 (2)-(4)，直到最顶层的事件类数目达到了一个预定的数值为止。该算法的时间复杂度为 $O(mn)$，其中 n 是输入语料库中的文档数，m 是每个组的大小，并且 m, n 满足 $m \leqslant n$ 条件。另外，还能够通过重聚类的方法来减少初始化组的系统偏差，从而能够生成一个更加紧凑的聚类分布。

INCR 聚类算法则依次处理输入文本，然后依次扩大聚类的集合。如果该文档与某个事件类的中心之间的相似度高于预先选择的阈值，则新的文本会被加入到已经生成的相似度最高的事件类中；否则，该文档将被视为新事件类。除此之外，还引入“新事件”和“旧事件”标签，通过该标签可以判断当前新闻是否是该时间点的新型事件。通过调整阈值，也可以得到不同粒度级别的集合。还可以额外利用输入数据的动态特性和事件的时间属性这两项信息，以达到提升算法效果的目的。

对于在线抽取算法，时间窗口的引入能够限制前 m 个新闻事件文档。对于按照序列依次处理的现有文档，每个在时间窗口文档相似度得分都会被计算，如果在窗口中相似度得分低于一个预定的阈值，那么将一个“新事件”的标签赋予给这个新闻文档。该判定的可信度 $score$ 设置为如下:

$$\text{score}(x) = 1 - \max_{d_i \in window} \{sim(x, d_i)\} \tag{7.83}$$

其中 x 是当前的文档，d_i 是窗口中的第 i 个文档，$i = 1, 2, ..., m$. 为了进一步更平滑的方式来使用时间临近信息，可以进一步弱化时间的权重:

$$\text{score}(x) = 1 - \max_{d_i \in window} \{\frac{i}{m} sim(x, d_i)\}. \tag{7.84}$$

这些窗口策略通过牺牲了少量的召回率来获得了精度的提高。i/m 线性衰减时间窗口结果与均匀加权窗口相比，始终能够产生更好的结果。

在阈值检测部分, 文献 [34] 所采用的方法主要是通过两个特定的阈值来进行控制。这两个阈值其分别是聚类阈值 t_c 和新奇性阈值 t_n, 前者决定聚类结果的颗粒度, 对于回顾抽取非常重要; 后者是决定算法敏感度和事件新奇性, 而对于在线抽取来说是关键影响因素。当 $t_c \geqslant t_n$，$sim_{max}(x) = 1 - score(x)$，在线抽取规则可以定义为:

- 如果 $sim_{max}(x) > t_c$，那么设定标志为 OLC，然后把文档 x 添加到窗口中最相似的事件类中。
- 如果 $t_c \geqslant sim_{max}(x) > t_n$，然后将标志设定为 Old，然后将文档 x 设定为新的独立事件类。
- 如果 $t_n > sim_{max}(x)$，然后设定标志为 New，然后将文档 x 设定为新的独立事件类。

2. 基于神经隐变量的开放域事件抽取方法

Xiao 等人[35] 提出了一种引入了神经网络隐变量的无监督生成模型 ODEE，进行新闻文本的事件抽取。输入为一个新闻集合（包含相同事件的报道），输出为一系列事件，每个事件都包含一个触发词和一个该事件模式的事件论元列表。该模型提取无约束的事件类型，并从新闻集群中归纳出通用的事件模式，每个新闻集群都有一个来自全局参数化正态分布的隐事件类型向量，以及实体的文本冗余特征。该模型利用了 ELMo 预训练语言模型和可扩展的神经变分推理。

针对开放域事件抽取任务，该方法提出了三个神经隐变量模型，它们的复杂程度依次增加，如图7.19所示。模型中 S 表示槽位数，E 表示实体数，C 表示新闻集合数，V 表示中心词汇量，灰色圆圈是可观测变量，白色圆圈是隐变量。

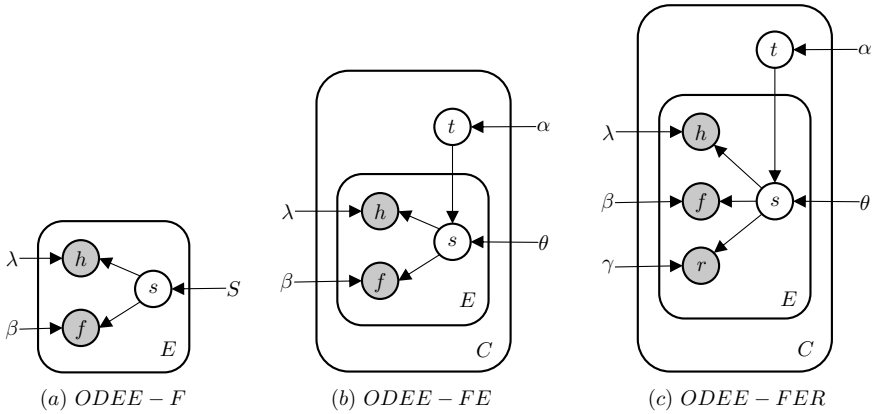


图 7.19　ODEE 方法三个神经隐变量模型结构图[35]

ODEE-F 模型如图7.19(a) 所示，给定一个语料库 N, 从 S 个槽的均匀分布中为每个实体 e 采样一个槽 s, 然后从多项分布中抽取一个中心词 (Head Word) h; 使用 ELMo 作为上下文编码器，得到连续特征向量 $\boldsymbol{f}$ （$\boldsymbol{f}$ 遵循多变量正态分布，其协方差矩阵是对角矩阵）。将 $\boldsymbol{f}$ 的 S 个不同正态分布的所有参数（协方差矩阵的均值向量和对角向量）标记为 $\boldsymbol{\beta} \in \mathbb{R}^{S\times(2n)}$，其中 n 表示 $\boldsymbol{f}$ 的维数，在逐行单纯形约束下槽分布概率矩阵 $\boldsymbol{\lambda} \in \mathbb{R}^{S\times V}$ 作为参数，其中 V 是中心词汇表大小。实体 e 的联合概率是:

$$p_{\lambda,\beta}(e) = p(s) \times p_{\lambda}(h \mid s) \times p_{\beta}(f \mid s) \tag{7.85}$$

ODEE-F 忽视了不同的事件可能有不同的槽分布，因此模型 ODEE-FE，如图7.19(b), 为每个新闻集从参数为 α 的全局正态分布中抽样一个潜在事件类型向量 $\boldsymbol{t}$, 然后使用 $\boldsymbol{t}$ 和参数 θ 的多层感知

器对相应的槽分布进行编码。新闻群 c 的联合概率是:

$$p_{\alpha,\beta,\theta,\lambda}(c) = p_\alpha(t) \times \prod_{e \in E_c} p_\theta(s \mid t) \times p_\lambda(h \mid s) \times p_\beta(f \mid s) \tag{7.86}$$

一个共指实体出现在新闻集群中的频率越高，它就越有可能是一个重要槽。除此之外，不同的新闻机构关注事件的不同方面，所以冗余的文本信息可以提供复杂的信息。因此，在模型 ODEE-FER 中，如图7.19(c) 所示，额外引入共指槽的归一化出现频率 r 作为观察到的隐变量。通常，一个新闻簇接收一个潜在事件类型向量 $\boldsymbol{t}$，其中每个实体 $e \in E_c$ 接收一个槽类型 s。具有中心词、冗余上下文特征和潜在事件类型的新闻簇的联合分布是:

$$p_{\alpha,\beta,\gamma,\theta,\lambda}(c) = p_\alpha(t) \times \prod_{e \in E_c} p_\theta(s \mid t) \times p_\lambda(h \mid s) \times p_\beta(f \mid s) \times p_\gamma(r \mid s) \tag{7.87}$$

在推理部分，考虑模型 ODEE-FER 处理的两个任务：(1) 学习参数和 (2) 在给定新闻集 c 的情况下执行推理以获得潜在变量 s 和 t 的后验分布。为简单起见，在模型 ODEE-FER 中将 $\boldsymbol{f}$ 和 $\boldsymbol{r}$ 连接起来作为新的观测特征向量 $\boldsymbol{f}'$，并将它们的参数合并为 $\boldsymbol{\beta}' \in \mathbb{R}^{S \times (2n+2)}$。将离散的潜变量 s 消去，获得对数似然的证据下界（Evidence Lower BOund，ELBO）:

$$\begin{aligned} \log p_{\alpha,\beta',\theta,\lambda}(c) &= \log \int_t \left[\prod_{e \in E_c} p_{\lambda,\theta}(h \mid t) p_{\beta',\theta}\left(f' \mid t\right) \right] p_\alpha(t) dt \\ &\geqslant \mathrm{ELBO}_c\left(\alpha, \beta', \theta, \lambda, \omega\right) \\ &= \mathbb{E}_{q_\omega(t)} \log p_{\beta',\theta,\lambda}(c \mid t) - D_{\mathrm{KL}}\left[q_\omega(t) \| p_\alpha(t)\right] \end{aligned} \tag{7.88}$$

其中 $D_{\mathrm{KL}}\left[q_\omega \| p_\alpha\right]$ 为 KL 散度。由于计算两个分布的 KL 散度非常困难，并且正态分布存在简单有效的重参数化技巧，因此选择 $q_\omega(t)$ 作为由 w 参数化的正态分布，由神经推理网络学习，如图7.20所示。

通过最大化下面的似然公式选择每个实体的槽:

$$\begin{aligned} p_{\beta',\theta,\lambda}(s \mid e, t) &\propto p_{\beta',\theta,\lambda}\left(s, h, f', t\right) \\ &= p_\theta(s \mid t) \times p_\lambda(h \mid s) \times p_{\beta'}\left(f' \mid s\right) \end{aligned} \tag{7.89}$$

最终新闻集合 c 组合成事件进行最终输出，还需要找到每个实体对应槽值的谓词。ODEE-FER 使用 Stanford Dependency Parser[36] 生成的词性标签和句法解析树，根据如下规则提取每个实体提及的中心词的谓词：(1) 如果一个中心词的支配者是 VB；或者 (2) 如果一个中心词的支配者是 NN 并且属于 WordNet 的 noun.ACT 或 noun.EVENT 范畴，那么该词为谓词。将相同共指链的实体提及的谓词合并为一个谓词集，对于集合中的每个谓词 v，找到其谓词集合包含 v 的实体，将这些实体视为由 v 触发的事件的论元。最终，对论元进行排序得到前 N 个开放域事件做为最终输出。

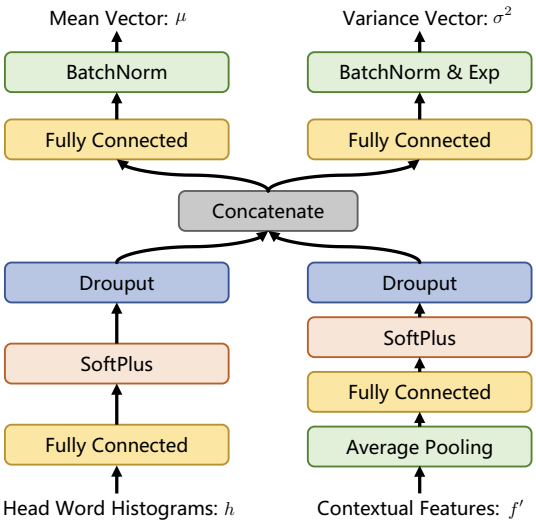


图 7.20 ODEE-FER 推理网络的框架[35]

图7.21给出了使用 ODEE-FER 进行开放事件抽取的示例。通过对比图7.21左边的文本内容和右侧所给出的事件信息，可以看到新闻被归纳为三个事件：“raise”、“report”和“predict”。不同的事件还产生了不同的槽位内容。

7.4.3 事件抽取评价方法

事件抽取的评价指标也采用统计机器学习算法评测中常用的指标进行对比，主要为精度（P）、召回率（R）、F 值，具体的计算方法如下：

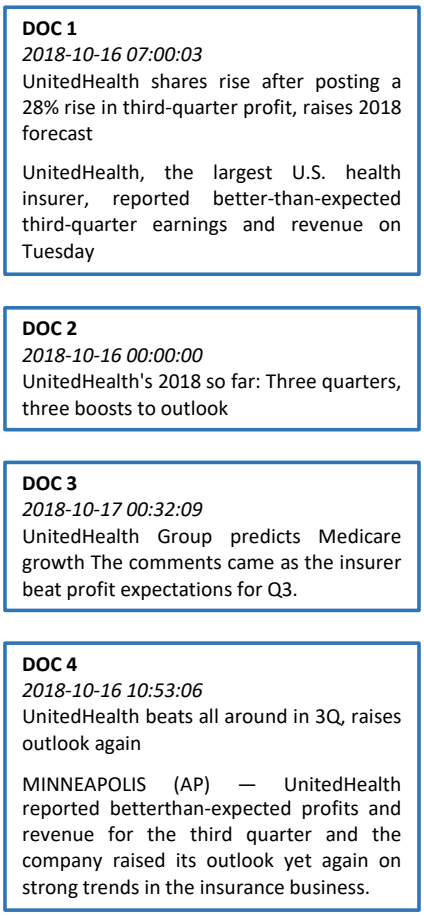
$$\text{精度（P）} = \frac{\text{正确抽取结果数}}{\text{抽取结果总数}} \times 100\% \tag{7.90}$$

$$\text{召回率（R）} = \frac{\text{正确抽取结果数}}{\text{需抽取结果总数}} \times 100\% \tag{7.91}$$

$$\text{F}_1 = \frac{2 \times P \times R}{P + R} \times 100\% \tag{7.92}$$

对于自动抽取系统或将事件抽取作为信息处理流水线的一部分时，应尽量提高 $F1$ 指标，以降低抽取错误造成后续步骤的错误累积；在有人工干预的事件抽取系统中，应在保证一定 $F1$ 指标的基础上，尽量提升召回率指标，以尽量确保抽取时不遗漏。

在评测基于流水线的事件抽取模型或系统时，有时还会使用上述指标对事件抽取中的各子任务分别进行评价，例如在一些论文中一般会同时汇报 TI（Trigger Identification，触发词识别）、TC



事件 1：

Tigger	raise
Agent	UnitedHealth, UnitedHealth shares
Patient	2018 forecast, better-than-expected profits, the insurance business
Time	the third quarter
Variation	28%

事件 2：

Tigger	report
Agent	UnitedHealth Group, the largest U.S. health insurer
Patient	better-than-expected third-quarter earnings
Time	Tuesday

事件 3：

Tigger	predict
Agent	UnitedHealth Group
Patient	Medicare growth

图 7.21　ODEE 生成的事件框架示例[35]

（Trigger Classification，触发词分类，即事件类型分类）、AI（Argument Identification，论元识别）、AC（Argument Classification，论元分类，即论元角色分类）四个子任务的精度、召回率以及 F1 值。

7.4.4 事件抽取语料库

事件抽取研究的发展同样离不开事件抽取相关评测集合和语料的不断推出。从最早的 MUC 语料库，再到目前使用最为广泛 ACE 语料库，以及中文 CEC 语料库，都极大的推动了事件抽取任务的不断进步。表7.4给出了常见事件抽取语料库的汇总。

语料库名称	事件类型数目	语言
ACE 事件语料库	8	中文、英文、阿拉伯文
MUC 语料库	4	英文
TDT 语料库	25	英文
CEC 语料库	5	中文

表 7.4 常见事件抽取语料库汇总

1. ACE 事件抽取语料库

ACE 事件语料库是目前事件抽取中最广泛使用的数据集之一，包含的事件具有复杂的结构和参数，涉及实体，时间和值。ACE 2005 事件语料库定义了 8 个事件类型和 33 子类型，每个事件子类型对应于一组参数角色。所有事件子类型共有 36 个参数角色，含中文、英文、阿拉伯语三种语言的语料。

2. MUC 事件抽取语料库

MUC 是最早产生支持事件共指任务的语料库。数据语料主要来自新闻语料，限定领域为飞机失事报道和航天器发射事件报道。MUC 评测中心围绕一个“场景”，根据关键事件类型和与它相关的各种角色定义。

3. TDT 事件抽取语料库

TDT 语料库来自于美国政府支持的 Topic Detectyion and Tracking 科研项目，其主要包含一个连续新闻流中的大量新闻，并对其进行了细分。整个语料库的所有标签都是人工手动标记的。它是由 15836 个发生在 1994 年 7 月 1 号到 1995 年 6 月 30 号之间的新闻事件组成的语料库。一半的新闻来自于路透社杂志，另一半来自于 CNN 多个广播新闻项目。整个语料库包含了 25 种事件，事件的定义仅仅给出的指示信息是发生的具体位置与具体时间。这些信息能够有效的将不同的事件区分开来。

4. CEC 中文事件抽取语料库

中文事件语料库（Chinese Event Corpus, CEC）由上海大学语义智能实验室构建，包含 CEC-1 和 CEC-2 两个语料库包。其中从互联网上收集了 5 类（地震、火灾、交通事故、恐怖袭击和食物中毒）突发事件的新闻报道作为生语料，然后再对生语料进行文本预处理、文本分析、事件标注以及一致性检查等处理，最后将标注结果保存到语料库中，CEC 合计 332 篇。与 ACE 和 TimeBank 语料库相比，CEC 语料库的规模虽然偏小，但是对事件和事件要素的标注却最为全面。

7.5 延伸阅读

尽管基于深度学习的信息抽取已经取得了显著的进步，但实际应用中场景的多元化使得信息抽取依然面临着诸多挑战，低资源、开放领域下模型的抽取能力、如何融入视觉、听觉等多种模

态来进一步提升信息抽取性能，以及如何改进框架将不同子任务统一建模，这些都值得进一步的探索。

(1) 高效的小样本学习能力。真实场景下的训练数据是十分有限的，这就要求模型具备从少量的样本中学习到实体、关系、事件的特征。目前在信息抽取任务中，常使用度量学习[37–39]、元学习[40–42]、迁移学习[43, 44]、融合领域知识[45–47] 等方式来提升模型从小样本中抽取信息的能力。随着基于 GPT3 等在内的超大规模预训练模型的 Prompt 学习范式受到研究者广泛关注，借助大规模预训练语言模型中蕴含的大量知识，仅利用几条或几十条样本作为训练集，在命名实体[48–50]、关系抽取[51–53] 等任务上，基于 Prompt 的方法也取得了不错的效果。

(2) 多模态信息融合。目前信息抽取主要针对的是纯文本数据，而常见的文档具有多样的布局且包含丰富的信息。此外很多文档、网页、社会媒体也是多以富文本的形式呈现，其中也包含大量的多模态信息。我们在 2018 年针对社会媒体多模态的命名实体[54] 论文也指出，在很多情况下仅依赖文本内容，无法完成准确的信息抽取任务。针对多模态信息抽取，研究人员从基于图对齐与图融合[55–57]、图卷积[58, 59]、结构化和半结构化页面结构[60, 61] 等方面进行了一些列研究。如何利用视觉、听觉、以及富文档信息，通过多模态信息补全文本中的缺失，是信息抽取的重要发展方向之一。

(3) 子任务统一建模。在本章中介绍的命名实体识别、关系抽取以及事件抽取任务都采用了不同类型的算法，关系抽取和事件抽取任务本身也会采用流水线方式组合多种算法。但是，随着“预训练 + 大规模多任务学习”这一范式所展现的学习能力，使得构建统一框架建模多种信息抽取任务成为可能。文献 [62] 即针对信息抽取设计了一种结构化抽取语言 (Structural Extraction Language, SEL)，采用 Seq2Seq 的生成式框架，并将命名实体识别、关系抽取、事件抽取这三个信息抽取任务的不同结构进行统一描述，使得模型针对不同任务输出一致的结构，实现了面向信息抽取的统一文本到结构生成框架 UIE（Universal Information Extraction）。文献 [63] 提出了基于 Prompt 的生成式方法，并构建了信息抽取任务容易框架。

7.6 习题

(1) 命名实体识别中有哪些解码方式？如何解决嵌套实体问题？

(2) 远程监督是关系抽取任务中自动标注训练数据的有效策略，但其过强的设定会产生错误标注，可以从哪些角度考虑来缓解远程监督引入的噪声问题？

(3) 试比较流水线式的关系抽取和联合关系抽取的优缺点。

(4) 限定域事件抽取的基本事件结构？

(5) 信息抽取目前还面临哪些挑战？如何解决开放域下的关系抽取问题？

参考文献

[1] Sarawagi S, Cohen W W. Semi-markov conditional random fields for information extraction[J]. Advances in neural information processing systems, 2004, 17.

[2] Yan H, Deng B, Li X, et al. Tener: adapting transformer encoder for named entity recognition[J]. arXiv preprint arXiv:1911.04474, 2019.

[3] Zhang Y, Yang J. Chinese ner using lattice lstm[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 1554-1564.

[4] Sohrab M G, Miwa M. Deep exhaustive model for nested named entity recognition[C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018: 2843-2849.

[5] Finkel J R, Manning C D. Nested named entity recognition[C]//Proceedings of the 2009 conference on empirical methods in natural language processing. 2009: 141-150.

[6] Xu M, Jiang H, Watcharawittayakul S. A local detection approach for named entity recognition and mention detection[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017: 1237-1247.

[7] Tan C, Qiu W, Chen M, et al. Boundary enhanced neural span classification for nested named entity recognition[C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 34. 2020: 9016-9023.

[8] Yan H, Gui T, Dai J, et al. A unified generative framework for various ner subtasks[C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2021: 5808-5822.

[9] Sang E T K, De Meulder F. Introduction to the conll-2003 shared task: Language-independent named entity recognition[C]//Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. 2003: 142-147.

[10] Weischedel R, Palmer M, Marcus M, et al. Ontonotes release 5.0 ldc 2013t19[J]. Linguistic Data Consortium, Philadelphia, PA, 2013, 23.

[11] Levow G A. The third international chinese language processing bakeoff: Word segmentation and named entity recognition[C]//Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing. 2006: 108-117.

[12] Peng N, Dredze M. Named entity recognition for chinese social media with jointly trained embeddings[C]//Proceedings of the 2015 conference on empirical methods in natural language processing. 2015: 548-554.

[13] Doddington G R, Mitchell A, Przybocki M A, et al. The automatic content extraction (ace) program-tasks, data, and evaluation.[C]//Lrec: volume 2. Lisbon, 2004: 837-840.

[14] Mani I, Hitzeman J, Richer J, et al. Ace 2005 english spatialml annotations[J]. Linguistic Data Consortium, Philadelphia, 2008.

[15] Kim J D, Ohta T, Tateisi Y, et al. Genia corpus—a semantically annotated corpus for bio-textmining [J]. Bioinformatics, 2003, 19(suppl_1):i180-i182.

[16] Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]//Proceedings of the 2008 ACM SIGMOD international conference on Management of data. 2008: 1247-1250.

[17] Zhang Y, Qi P, Manning C D. Graph convolution over pruned dependency trees improves relation extraction[C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 2205-2215. https://aclanthology.org/D18-1244. DOI: 10.18653/v1/D18-1244.

[18] Zeng D, Liu K, Chen Y, et al. Distant supervision for relation extraction via piecewise convolutional neural networks[C]//EMNLP. 2015.

[19] Lin Y, Shen S, Liu Z, et al. Neural relation extraction with selective attention over instances[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: 2124-2133. https://aclanthology.org/P16-1200. DOI: 10.18653/v1/P16-1200.

[20] Etzioni O, Banko M, Soderland S, et al. Open information extraction from the web[J/OL]. Commun. ACM, 2008, 51(12):68–74. https://doi.org/10.1145/1409360.1409378.

[21] Downey D, Etzioni O, Soderland S. A probabilistic model of redundancy in information extraction [C]//Proceedings of the 19th international joint conference on Artificial intelligence. 2005: 1034-1041.

[22] Fader A, Soderland S, Etzioni O. Identifying relations for open information extraction[C/OL]// Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Edinburgh, Scotland, UK.: Association for Computational Linguistics, 2011: 1535-1545. https://aclanthology.org/D11-1142.

[23] Hu X, Wen L, Xu Y, et al. Selfore: Self-supervised relational feature learning for open relation extraction[C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020: 3673-3682.

[24] Zhao J, Gui T, Zhang Q, et al. A relation-oriented clustering method for open relation extraction[C]// Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021: 9707-9718.

[25] Hendrickx I, Kim S N, Kozareva Z, et al. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals[C/OL]//Proceedings of the 5th International Workshop on Semantic Evaluation. Uppsala, Sweden: Association for Computational Linguistics, 2010: 33-38. https://aclanthology.org/S10-1006.

[26] Zhang Y, Zhong V, Chen D, et al. Position-aware attention and supervised data improve slot filling [C/OL]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017). 2017: 35-45. https://nlp.stanford.edu/pubs/zhang2017tacred.pdf.

[27] Han X, Zhu H, Yu P, et al. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation[C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018: 4803-4809.

[28] Riedel S, Yao L, McCallum A. Modeling relations and their mentions without labeled text[C]//Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2010: 148-163.

[29] Ahn D. The stages of event extraction[C]//Proceedings of the Workshop on Annotating and Reasoning about Time and Events. 2006: 1-8.

[30] Daelemans W, Zavrel J, Van Der Sloot K, et al. Timbl: Tilburg memory-based learner[J]. Tilburg University, 2004.

[31] Daumé III H. Notes on cg and lm-bfgs optimization of logistic regression[J]. Paper available at http://pub. hal3. name# daume04cg-bfgs, implementation available at http://hal3. name/megam, 2004, 198:282.

[32] Chen Y, Xu L, Liu K, et al. Event extraction via dynamic multi-pooling convolutional neural networks [C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015: 167-176.

[33] Nguyen T H, Cho K, Grishman R. Joint event extraction via recurrent neural networks[C]// Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016: 300-309.

[34] Yang Y, Pierce T, Carbonell J. A study of retrospective and on-line event detection[C]//Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. 1998: 28-36.

[35] Liu X, Huang H Y, Zhang Y. Open domain event extraction using neural latent variable models[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 2860-2871.

[36] Klein D, Manning C D. Accurate unlexicalized parsing[C]//Proceedings of the 41st annual meeting of the association for computational linguistics. 2003: 423-430.

[37] Fritzler A, Logacheva V, Kretov M. Few-shot classification in named entity recognition task[C]// Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. 2019: 993-1000.

[38] Gao F, Cai L, Yang Z, et al. Multi-distance metric network for few-shot learning[J]. International Journal of Machine Learning and Cybernetics, 2022:1-12.

[39] Huang Y, He K, Wang Y, et al. Copner: Contrastive learning with prompt guiding for few-shot named entity recognition[C]//Proceedings of the 29th International Conference on Computational Linguistics. 2022: 2515-2527.

[40] Li J, Chiu B, Feng S, et al. Few-shot named entity recognition via meta-learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2020.

[41] de Lichy C, Glaude H, Campbell W. Meta-learning for few-shot named entity recognition[C]// Proceedings of the 1st Workshop on Meta Learning and Its Applications to Natural Language Processing. 2021: 44-58.

[42] Huang J, Li C, Subudhi K, et al. Few-shot named entity recognition: An empirical baseline study [C]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021: 10408-10423.

[43] Rahimi A, Li Y, Cohn T. Massively multilingual transfer for ner[C]//ACL 2019-57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference. Association for Computational Linguistics-ACL, 2019: 151-164.

[44] Jia C, Zhang Y. Multi-cell compositional lstm for ner domain adaptation[C]//Proceedings of the 58th annual meeting of the association for computational linguistics. 2020: 5906-5917.

[45] Ma R, Peng M, Zhang Q, et al. Simplify the usage of lexicon in chinese ner[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 5951-5960.

[46] Peng M, Ma R, Zhang Q, et al. Toward recognizing more entity types in ner: an efficient implementation using only entity lexicons[C]//Findings of the Association for Computational Linguistics: EMNLP 2020. 2020: 678-688.

[47] Wu S, Song X, Feng Z. Mect: Multi-metadata embedding based cross-transformer for chinese named entity recognition[C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2021: 1529-1539.

[48] Chen X, Li L, Deng S, et al. Lightner: A lightweight tuning paradigm for low-resource ner via pluggable prompting[C]//Proceedings of the 29th International Conference on Computational Linguistics. 2022: 2374-2387.

[49] Ma R, Zhou X, Gui T, et al. Template-free prompt tuning for few-shot NER[C/OL]//Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Seattle, United States: Association for Computational Linguistics, 2022: 5721-5732. https://aclanthology.org/2022.naacl-main.420. DOI: 10.18653/v1/2022.naacl-main.420.

[50] Lai P, Ye F, Zhang L, et al. Pcbert: Parent and child bert for chinese few-shot ner[C]//Proceedings of the 29th International Conference on Computational Linguistics. 2022: 2199-2209.

[51] Chen X, Zhang N, Xie X, et al. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction[C]//Proceedings of the ACM Web Conference 2022. 2022: 2778-2788.

[52] Zhang H, Liang B, Yang M, et al. Prompt-based prototypical framework for continual relation extraction[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2022, 30:2801-2813.

[53] Yang S, Song D. Fpc: Fine-tuning with prompt curriculum for relation extraction[C]//Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing. 2022: 1065-1077.

[54] Zhang Q, Fu J, Liu X, et al. Adaptive co-attention network for named entity recognition in tweets [C]//Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[55] Yu J, Jiang J, Xia R. Entity-sensitive attention and fusion network for entity-level multimodal sentiment classification[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2019, 28:429-439.

[56] Zheng C, Feng J, Fu Z, et al. Multimodal relation extraction with efficient graph alignment[C]// Proceedings of the 29th ACM International Conference on Multimedia. 2021: 5298-5306.

[57] Zhang D, Wei S, Li S, et al. Multi-modal graph fusion for named entity recognition with targeted visual guidance[C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 35. 2021: 14347-14355.

[58] Liu X, Gao F, Zhang Q, et al. Graph convolution for multimodal information extraction from visually rich documents[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers). 2019: 32-39.

[59] Yu W, Lu N, Qi X, et al. Pick: processing key information extraction from documents using improved graph learning-convolutional networks[C]//2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021: 4363-4370.

[60] Lockard C, Shiralkar P, Dong X L, et al. Zeroshotceres: Zero-shot relation extraction from semi-structured webpages[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 8105-8117.

[61] Li Y, Qian Y, Yu Y, et al. Structext: Structured text understanding with multi-modal transformers [C]//Proceedings of the 29th ACM International Conference on Multimedia. 2021: 1912-1920.

[62] Lu Y, Liu Q, Dai D, et al. Unified structure generation for universal information extraction[C/OL]// Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume

1: Long Papers). Dublin, Ireland: Association for Computational Linguistics, 2022: 5755-5772. https://aclanthology.org/2022.acl-long.395.

[63] Kan Z, Feng L, Yin Z, et al. A unified generative framework based on prompt learning for various information extraction tasks[J]. arXiv preprint arXiv:2209.11570, 2022.

索引