



自然语言处理导论

张奇 桂韬 黄萱菁

2023 年 1 月 4 日

数学符号

数与数组

α	标量
α	向量
A	矩阵
\mathbf{A}	张量
I_n	n 行 n 列单位矩阵
v_w	单词 w 的分布式向量表示
e_w	单词 w 的独热向量表示: $[0,0,\dots,1,0,\dots,0]$, w 下标处元素为 1

索引 |

α_i	向量 α 中索引 i 处的元素
α_{-i}	向量 α 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 A 中第 i 行、第 j 列处的元素
$A_{i:}$	矩阵 A 中第 i 行
$A_{:j}$	矩阵 A 中第 j 列
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
$0, 1, \dots, n$	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

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

微积分

$\frac{dy}{dx}$	y 对 x 的导数
$\frac{\partial y}{\partial x}$	y 对 x 的偏导数
$\nabla_{\mathbf{x}} y$	y 对向量 \mathbf{x} 的梯度
$\nabla_{\mathbf{X}} y$	y 对矩阵 \mathbf{X} 的梯度
$\nabla_{\mathbf{X}} y$	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
$\text{Var}(f(x))$	$f(x)$ 在分布 $P(x)$ 下的方差
$\text{Cov}(f(x), g(x))$	$f(x)$ 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
$\mathbf{x}^{(i)}$	数据集中第 i 个样本（输入）
$\mathbf{y}^{(i)}$ 或 $y^{(i)}$	第 i 个样本 $\mathbf{x}^{(i)}$ 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
$f(\mathbf{x}; \boldsymbol{\theta})$	由参数 $\boldsymbol{\theta}$ 定义的关于 \mathbf{x} 的函数（也可以直接写作 $f(\mathbf{x})$, 省略 $\boldsymbol{\theta}$ ）
$\log x$	x 的自然对数函数
$\sigma(x)$	Sigmoid 函数 $\frac{1}{1 + \exp(-x)}$
$\ \mathbf{x}\ _p$	\mathbf{x} 的 L^p 范数
$\ \mathbf{x}\ $	\mathbf{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 $\mathbf{x} = x_1, x_2, \dots, x_n$ 中第 i 个单词 x_i 的词向量 \mathbf{v}_{x_i}
- 损失函数 \mathcal{L} 为负对数似然函数： $\mathcal{L}(\boldsymbol{\theta}) = -\sum_{(x,y)} \log P(y|x_1 \dots x_n)$
- 算法的空间复杂度为 $\mathcal{O}(mn)$

目 录

5 篇章分析	1
5.1 篇章理论概述	1
5.1.1 篇章的衔接	2
5.1.2 篇章的连贯	4
5.1.3 篇章的结构	5
5.2 话语分割	9
5.2.1 基于词汇句法树的统计话语分割	10
5.2.2 基于循环神经网络的话语分割	11
5.3 篇章结构分析	13
5.3.1 修辞结构篇章分析	13
5.3.2 浅层篇章分析	17
5.4 指代消解	23
5.4.1 基于表述对的指代消解	24
5.4.2 基于表述排序的指代消解	27
5.4.3 基于实体的指代消解	31
5.5 延伸阅读	35
5.6 习题	35

5. 篇章分析

到目前为止，本书中讨论的都是词语或句子层面的语言现象。然而，语言通常并不是由独立无关的句子组成，而是由搭配在一起、具有一定结构的连贯的（Coherent）句子集合组成。我们将这样的句子集合称为篇章（Discourse）。篇章分析的目的是从整体上理解篇章，其中最重要的是对篇章的连贯性和衔接性进行分析。连贯性是将真正的篇章区别于无关、随机的句子集合的重要性质，而衔接性帮助我们分析和理解篇章的结构，包括其名词、代词之间的指代关系等。篇章分析在自然语言处理中具有非常重要的作用，是摘要生成、阅读理解等篇章级别任务的必要环节。

本章首先介绍篇章分析的基本概念，在此基础上介绍篇章分析的三个子任务：话语分割、话语分析和指代消解的主要算法和语料库。

5.1 篇章理论概述

篇章语言学是在二十世纪五十年代以后发展起来的一门新兴学科。传统语言学通常以句子本身及其组成部分为研究对象。但是随着语言学研究的进展，人们发现句子在不同的上下文和语境中也可以有不同的意义或者具有不同交际功能，多个合乎句法的句子也并不是随意堆在一起就能构成一个合格的语篇。越来越多的语言学家开始认识到语言研究应该超越句子层次，句子的组合受到语法以外的规则的制约。Harris (1952) 在《话语分析》(Discourse Analysis) 一书中首次提出了“话语分析”这一术语。W. Weinrich 在 1967 年首次提出了“篇章语言学”这一概念，认为任何语言学研究都应该以语篇为描述框架。

篇章（Discourse）也称语篇，是指由一系列连续的语段或句子组成的整体，是语言运用或交际的基本单位。篇章的形式是多种多样的，既包含新闻、小说、论文、报告又包含警示标语、交通标识等。像“禁止通行”这样的警示语以及“停！”之类的有意义的语言单位都可以看作是一个篇章。但是，并不是所有大于句子的单位都可以组成一个合格的篇章。Beaugrande 和 Dressler 在 1981 年所著的《篇章语言学导论》(Introduction to text linguistics)^[1] 一书中指出，一个合格的篇章需要满足七个标准：衔接 (cohesion)、连贯 (coherence)、意图性 (intentionality)、可接受性 (acceptability)、信息性 (informativity)、情景性 (situationality) 和互文性 (intertextuality)。由此，可以看到篇章与孤立句子的主要区别在于：篇章是由句子组成的前后连贯的、有主题的统一整体。篇章所呈现

的特定结构，不仅包含音、词和句法等表层结构上，也体现在语义连贯的深层结构上。需要注意的是，由于篇章的类型和特点千差万别，一个合格的篇章并不一定完全满足上述所有七个标准。

本节中针对篇章分析中最重要三个方面：衔接、连贯和组织对篇章语言学理论进行简略介绍。

5.1.1 篇章的衔接

衔接（Cohesion）是指篇章中的某一语言成分需要依赖另一语言成分进行解释^[2]。衔接是一种语义关系，使得篇章各组成部分在语义上相互联系，关系紧凑。衔接也被作为语篇连贯性的必要条件之一，在语篇中体现为词汇衔接和语法衔接。词汇衔接包括重述（reiteration）、搭配（collocation）等衔接手段。语法衔接包括照应（reference）、替代（substitution）、省略（ellipsis）、连接（conjunction）等衔接手段。

1. 词汇衔接

重述关系是通过词的重复、同义词或近义词、反义词、上下位词等词汇手段形成的篇章衔接关系。例如：

- (1) 苏州园林据说有一百多处，我到过的不过十多处。其他地方的园林我也到过一些。倘若要我说说总的印象，我觉得苏州园林是我国各地园林的标本，各地园林或多或少都受到苏州园林的影响。因此，谁如果要鉴赏我国的园林，苏州园林就不该错过。

（《苏州园林》）

- (2) 那棵树立在那条路边上已经很久很久了。当那路还只是一条泥泞的小径时，它就立在那里；当路上驶过第一辆汽车之前，它就立在那里；当这一带只有稀稀落落几处老式平房时，它就立在那里。

（《那树》）

在上述例子（1）中“苏州园林”出现了四次，“我国的园林”、“各地园林”等则通过上下位关系进行衔接。例子（2）中，“它就立在那里”出现了三次，与“已经很久很久了”也形成了近义重述关系。

搭配关系是指词的共现关系，包括一个词组或者一个句子内部的词之间的组合关系，也包括句子间或段落间的词的习惯性共现。例如：

有一天早上，撒了三次网，什么都没捞着，他很不高兴。第四次把网拉拢来的时候，他觉得太重了，简直拉不动。他就脱了衣服跳下水去，把网拖上岸来。打开网一看，发现网里有一个胆形的黄铜瓶，瓶口用锡封着，锡上盖着所罗门的印。

（《渔夫的故事》）

上述例子中，围绕“撒网”展开，形成了一个与撒网打鱼相关的动词链，“撒-捞-拉-拖”够成了词汇衔接。在同一个篇章中，词汇之间通过语义联想构成衔接关系。

2. 语法衔接

照应是指篇章中一个语言成分与另一可以与之相互解释的成分之间的关系，即一个成分作为另一个成分的参照点。例如：

那只最后从蛋壳里爬出来的小鸭是那么丑陋，他处处挨啄，被排挤，被讪笑，不仅在鸭群中是如此，连在鸡群中也是这样。

(《丑小鸭》)

这里代词“他”的确切含义是由它所指的对象决定的。本例中“他”是指“最后从蛋壳里爬出来的小鸭”。

照应性 (Phoricity) 是语言交际过程中一个普遍现象，用来指代篇章中的实体、概念或事件。照应可以分为两种：外指 (Exophora) 和内指 (Endophora)。外指照应是指篇章中的某个成分的参照点不在篇章本身，而是在语境中。内指照应是指语言成分的参照点在篇章上下文中。内指照应又可以进一步细分为回指 (Anaphora) 和下指 (Cataphora)。回指照应是指所指对象位于上文；下指照应是指所指对象位于下文。在词汇语法层面，照应还可以分为人称照应、指示照应和比较照应，分别是指用人称代词、指示代词以及表示比较的形容词副词所表示的照应关系。照应在篇章中具有重要的作用，可以使篇章在结构上更加紧凑，同时在修辞上达到言简意赅的效果。

替代是指用替代形式来取代上文中的某一成分。在篇章中，由于替代形式的意义必须通过所替代的成分才能获取，因而替代起到了衔接的作用，使得替代成分和替代对象所属句子紧密连接。从语法和修辞角度，替代也是避免重复的一种重要语言手段。替代可以进一步细分为名词性替代 (Nominal Substitution)、动词性替代 (Verbal Substitution) 和小句性替代 (Clausal Substitution)。例如：

各式各样的球鞋像装在万花筒里，在她面前转开了：白色的，蓝色的，高筒的，矮帮的，白色带红边的，白色带蓝边的。

(《人到中年》)

上例中“白色的，蓝色的，高筒的，矮帮的，白色带红边的，白色带蓝边”与上一句中“球鞋”构成了紧密的衔接关系，“的”替代了上文中的“球鞋”，属于名词性替代。汉语中“做”和“干”经常用于动词性的替代。“这样”、“这么”等经常用于小句性替代。

省略是指将语言结构中某个成分在句子中去除。虽然省略结构在语法层面不完整，但是并不是不可理解的，并且表达更加精炼。由于省略成分需要从上下文中获取，也使得省略成为了常见的语法衔接手段。例如：

雨是最寻常的，一下就是三两天。可别恼。看，像牛毛，像花针，像细丝，密密地斜织着，人家屋顶上全笼着一层薄烟。

(《春》)

上例中“像牛毛，像花针，像细丝”前都省略了“雨”，但是很容易理解并且语篇前后衔接，结构紧凑。省略也可以分为名词性省略 (nominal ellipsis)、动词性省略 (verbal ellipsis) 以及小句性省略 (clausal ellipsis)。

连接是通过连接成分体现篇章中逻辑关系。从逻辑语义关系类型上，可以细分为三大类：详述 (Elaboration)、延伸 (Extension) 和增强 (Enhancement)。详述是对上文内容进一步说明、评论或解释，主要包括同位语和阐明两种情况。延伸是从正面或反面增加新的陈述，包括添加、转折、变

换等类型。增强则是指补充额外必要信息，达到加强语义并使其更加完整，包括时空、方式、因果与条件、话题等条件。例如：

不必说碧绿的菜畦，光滑的石井栏，高大的皂荚树，紫红的桑葚；也不必说鸣蝉在树叶里长吟，肥胖的黄蜂伏在菜花上，轻捷的叫天子（云雀）忽然从草间直窜向云霄里去了。单是周围的短短的泥墙根一带，就有无限趣味。

（《从百草园到三味书屋》）

上例中通过“不必说”、“也不必说”、“单是”层层递进，使得句子之间紧密连接。

5.1.2 篇章的连贯

连贯（Coherence）是指篇章在语义、功能和心理上构成一个整体，围绕同一个主题或意图展开^[3]。连贯性（coherent）是衡量篇章质量的重要指标，只有连贯的句子集合才能够形成篇章。这也是篇章与无关的、随机的句子集合区分开的最主要因素。篇章应该同时具有局部连贯性（Local Coherent）和整体连贯性（Global Coherent）。局部连贯性是在微观层面，篇章中前后相连的命题在语义上的联系。整体连贯性是在宏观层面，篇章中的所有命题与篇章主题之间的联系。

篇章局部连贯通常是通过话语序列的语义结构实现。一般来说，话语序列的语义结构表现可以分为外延的（Extensional）和内涵的（Intensional）两种类型。外延的语义结构表示话语序列所表达的事态与真实世界的排序顺序相对应；内涵的语义结构表示话语序列所表达的事态在真实世界中找不到对应。例如：

- (1) 他点了一份外卖。
- (2) 外卖很快就送到了。

上例中，(1) 表示 (2) 的条件，并且与真实世界中事件存在对应关系，属于外延语义结构。为了达到话语序列在语义上的连贯性，需要与现实世界中的自然顺序相对应，也就是说如果把 (1) 和 (2) 的顺序颠倒过来，那么该话语序列就不再具有语义上的连贯性。

除了现实世界中自然顺序的限制，话语序列的语义结构还受到人们普遍认知规律的制约。人们认识和描述客观世界时通常遵循从一般到特殊、从整体到局部、从大到小，从集合到子集的认知模式。例如：

单是周围的短短的泥墙根一带，就有无限趣味。油蛉在这里低唱，蟋蟀们在这里弹琴。翻开断砖来，有时会遇见蜈蚣；还有斑蝥，倘若用手指按住它的脊梁，便会啪的一声，从后窍喷出一阵烟雾。

（《从百草园到三味书屋》）

上例符合整体到局部的排列顺序，从对百草园的“无限趣味”开始，再以局部的细节展开，详细描写了由“油蛉”、“蟋蟀”、“斑蝥”所带来的乐趣。这样的顺序与人们一般的认知规律和感知顺序相符合，从而也就更容易让人接受。

局部连贯说明篇章中相邻句子存在联系，但是仅有局部连贯是不够的，篇章在整体上还需要

围绕一个主题展开，既需要具有整体连贯性。整体连贯性对篇章中句子之间的联系施加宏观制约。例如：

(1) 对于一个在北平住惯的人，像我，冬天要是不刮风，便觉得是奇迹；济南的冬天是没有风声的。对于一个刚由伦敦回来的人，像我，冬天要能看得见日光，便觉得是怪事；济南的冬天是响晴的。自然，在热带的地方，日光永远是那么毒，响亮的天气，反有点儿叫人害怕。可是，在北方的冬天，而能有温晴的天气，济南真得算个宝地。

(《济南的冬天》)

(2) 母亲还从来没有一次给过我这么多钱。我也从来没有向母亲一次要过这么多钱。我来到母亲工作的地方，呆呆地将那些母亲扫视一遍，却没有发现我的母亲。背直起来了，我的母亲。转过身来了，我的母亲。褐色的口罩上方，一对眼神疲惫的眼睛吃惊地望着我，我的母亲。

上例中，(1) 在微观层面上前后承接，句子之间的逻辑关系清晰，宏观层面围绕“济南的冬天”主题开展。例(2) 的局部上是连贯的，但是宏观层面上缺乏主题，从而缺乏整体连贯。

篇章的连贯是一个复杂的现象，有些现象不能完全从语义角度解释话语序列的连贯性，还需要从语用角度以及认知角度进行讨论。围绕语篇的连贯也有很多理论和方法，包括从关联理论角度对微观层面连贯性研究，利用修辞结构理论 (Rhetorical Structure Theory) 进行语篇连贯性研究，运用图式理论 (Schema Theory) 的连贯性研究，基于语篇策略 (Discourse Strategy) 的连贯性研究等。研究篇章的连贯性是自然语言处理中的重要问题，对文本摘要、阅读理解、机器翻译等篇章级别的任务都具有重要的作用。

5.1.3 篇章的结构

篇章同时具有线性结构和等级结构。篇章中的句子按照一定的线性规则排列在一起，因此篇章是线性的。同时，句子的组合可以构成更大的语言单位，因此篇章又是具有等级结构的。例如：

[1] 没有春节不是流动的，也没有春节不是走动的。[2] 这是以往中国人过春节的常态，热热闹闹、走亲串户、朋友相聚，动起来的春节被视为祥和、欢乐的时节。

[2] 然而，这个春节，真的不一样。[3] 一个现实原因就是，新型冠状病毒引发的疫情还在持续，全国人民为此揪心。[4] 应该以什么样的状态与心态，过好这个春节，值得我们细细思量。[5] 春节的流动、拜年的走动、庙会的人头攒动，这些人们已经习惯了的过年方式，在这些日子里恐怕需要改一改了。

[6] 此时，“动”的年节莫若“静”的岁月。[7] 人们越是大规模流动，越是大范围聚集，越容易增加疾病传染的概率。[8] 走动起来还是宅上一宅，理性人不难看透其中的得失，既为人也为己。[9] 事实上，不走动也能过好年。[11] 技术发达了，信息拜年、视频祝福、在线聚会，都不失为一种时尚，那些以往通过面对面完成的新春祝福，借助云端就能迅速直抵耳畔、身边，过年礼仪一样也缺不了。

[12] 此时，“动”的脚步莫若“静”的心意。[13] 在抗击疫情的最前沿，各条战线上的

“勇士”都已经动起来了，他们为了更多人的生命安全，以这样一种方式过了个“动”的年，是真正的大无畏。^[14]相反，对普通人来说，如无特殊情况，宜静不宜动，什么自驾跨城回家、什么一定上门拜年、什么提前安排好的聚会等等，都不妨在冷静且理性地审视下做个宅男宅女，不远行、不扎堆、少聚会。^[15]现在，最好的祝福是以你我的安全距离为彼此送上健康祝福，最大的心意是以你我的实际行动护佑早日战胜疫情。

(《这个春节，我们以“静”致敬》)

上例中，句子[1][2]组成了引论，句子[2]-[5]给出了论点，句子[6]-[11]组成了第一个分论点，其中句子[6]是提出分论点，句子[7]-[11]是论据，句子[12]-[15]组成了第二个分论点，句子[12]是提出分论点，句子[13]-[15]是论据。

本节中将介绍三种常见的篇章结构的表示方法：超级结构（Superstructure）、修辞结构理论（Rhetorical Structure Theory）和语篇模式（Textual Pattern）。

1. 篇章超级结构

篇章超级结构（Superstructure）是采用规约化图式结构来表示篇章宏观内容组织形式的一种形式结构。只涉及篇章内容的组织方法，与篇章所表达的具体内容没有直接关系。不同类型的语篇往往具有不同的超级结构。比如，科技论文通常包含标题、摘要、引言、相关工作、方法介绍、实验、结论、参考文献等成分组成。

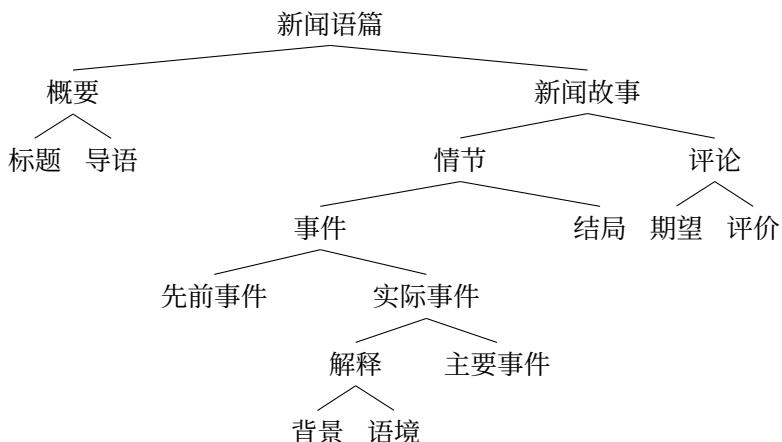


图 5.1 新闻篇章超级结构示例^[4]

新闻篇章的超级结构如图5.1所示。新闻报道一般由概述、故事和结局等结构要素组成。概述包括标题和导语，故事包括情节和背景，结局包括评论和结论^[4]。篇章超级结构提供了组织相关类型语篇的基础纲要和架构。在具体的篇章中，结构也具有一定的灵活性，并不是所有的结构成分都要存在，结构成分的位置也是不固定的。

2. 修辞结构理论

修辞结构理论（Rhetorical Structure Theory, 简称 RST）是 Mann 和 Thompson 于 1987 年提出的一种通过描述篇章各个组成部分之间的修辞关系来分析篇章结构的理论^[5]。修辞结构理论将修辞关系定义在两个或多个文本单元（Text Span）之间。文本单元又称基本篇章单元（Elementary Discourse Unit, EDU），有两种主要类型：核心（nucleus）和辅助（satellite）。核心单元是篇章中最重要的部分，表达作者的核心意图，并且具有相对完整的语义，能够独立解释；辅助单元则较少表达作者的核心意图，用于传达支撑其他信息，补充说明核心单元，通常只有在与核心单元关联时才能够被解释。修辞关系通常定义在核心单元和辅助单元之间，也有少部分修辞关系定义在两个或多个核心单元之间。例如：

[1] 这个草莓真的好吃，[2] 我吃了一大盆。

在上例中，小句 [1] 是一个陈述或者判断，小句 [2] 则为这一判断提供了证据。小句 [1] 是核心单元，小句 [2] 是辅助单元，两个小句之间构成证据关系（evidence）。该句可以使用如图5.2所示的“证据”关系的图示表示。

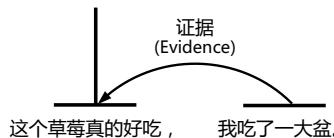


图 5.2 “证据”关系图示样例

根据修辞结构理论，篇章中存在多种多样的修辞结构关系，并且随着研究的不断深入，研究人员也在不断地对修辞关系进行补充。根据 Mann 和 Thompson^[5] 的定义，篇章中的修辞关系主要包括两种类型：1) 不对称性的核心-辅助关系（nucleus-satellite relation），也称“单核”关系；2) 无主次之分的“多核心关系”（multinuclear relation）。图5.3中给出了修辞结构理论中五种图示类型。竖线指示出核心单元，弧线连接具有关系的单元，关系的名称标注在连线上。环境（circumstance）关系是单核心关系，弧线箭头指向核心单元。对比（contrast）关系总是两个核心单元。序列（sequence）关系则可以具有多个连续的单元，相邻的两个单元之间构成序列关系。联合（joint）关系也可以具有多个单元，这些单元一起构成该关系。

篇章修辞关系中绝大多数都是核心-辅助关系，包括：对立（antithesis）、动机（motivation）、背景（background）、析取（otherwise）、意图（purpose）、总结（summary）、评价（evaluation）、证据（evidence）、使能（enablement）等。无主次修辞关系主要包含对比（contrast）、联合（joint）、列举（list）和序列（sequence）。此外，篇章的修辞结构在总体上表现为等级结构，连贯的篇章可以由不同层次的修辞关系组织成层次结构，从而形成一个修辞关系树。图5.4给出了根据修辞结构理论构成的一个修辞结构关系树样例。

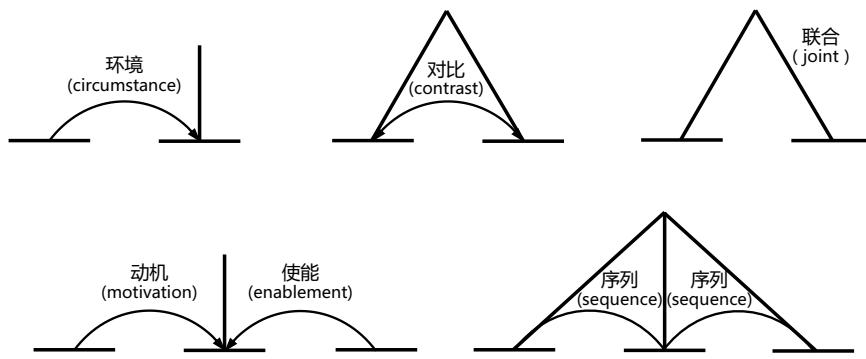


图 5.3 修辞结构理论中关系图示类型

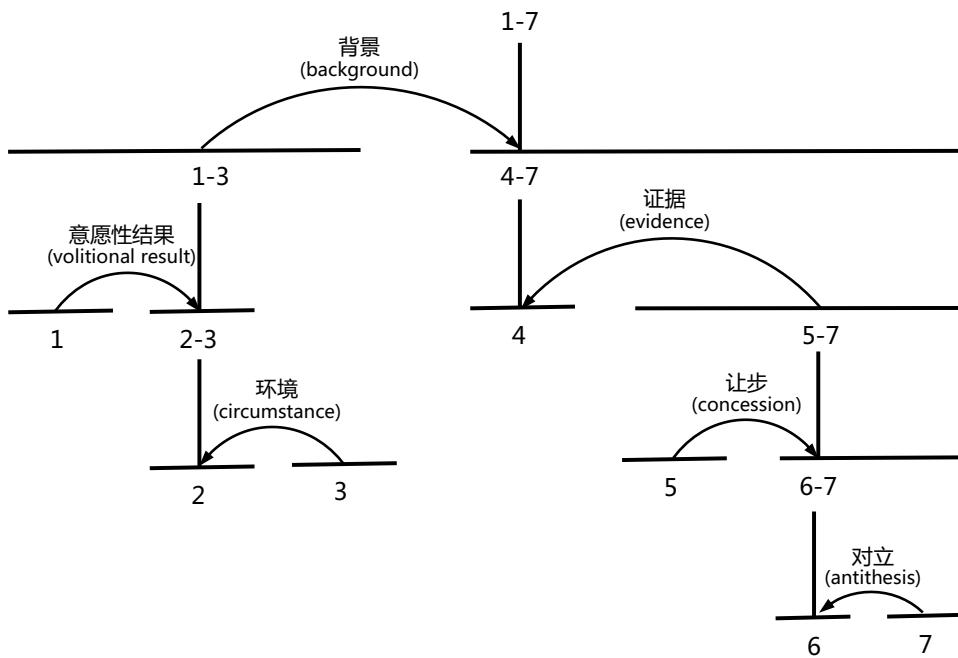


图 5.4 修辞结构关系树样例

3. 语篇模式

语篇模式（Textual Pattern）是指人们长期积累并根据经验形成的一些程式化的语篇组织形式或策略^[6]。语篇模式是在一定的文化中形成的，因此往往带有不同文化积淀的内涵和文化规约性。语

言学家总结出了“问题-解决”(Problem-Solution)、“概括-具体”(General-Specific)等英语中常见的语篇模式。语篇模式与小句关系之间存在着密切的联系，小句通过组合形成逻辑序列关系或匹配关系，通过这些关系小句又组合为更大的语篇单位。语篇模式与具体的篇章内容通常没有直接的联系，但是每种语篇模式通常都具有特定的词汇标记。

以问题-解决模式为例，该模式通常由四个部分构成：情景、问题、反应、评价。这个过程也符合人们通常的认知模式。例如：

[1] 长征五号遥三运载火箭 27 日晚在海南文昌一飞冲天，将实践二十号卫星成功送入太空预定轨道。[2] “胖五”也以实际行动，诠释着中国俗话所说“哪里跌倒，就要从哪里爬起来”的坚持与坚韧。[3]2017 年 7 月长征五号遥二火箭因发动机故障发射失利。[4] 科研人员历经两年多的艰苦攻关、连续奋战，进行大量地面试验，完成遥二失利故障归零和遥三火箭各项工作，还采取一系列改进优化措施，切实提升火箭飞行任务可靠性。[5] 长征五号遥三火箭在此背景下成功发射，对研制团队直面挑战、发现问题、解决问题的心理能力建设也是一次巨大考验，也为航天人才特别是青年人才树立起不怕失败、敢于挑战、勇于拼搏的榜样力量。

(1)

《述评：长征五号“王者归来”意味着什么》

上例中 [1][2] 句描述了情景，句子 [3] 说明了问题，[4] 句给了反应和解决问题的方法，最后句子 [5] 给出了评价。语篇模式如图 5.5 所示：

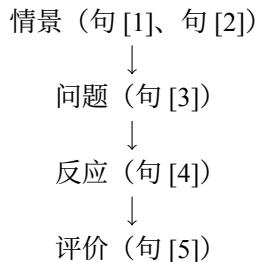
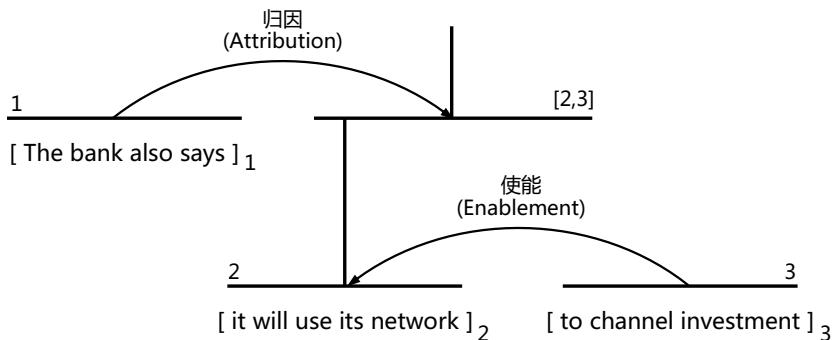


图 5.5 《述评：长征五号“王者归来”意味着什么》语篇模式示例

5.2 话语分割

根据修辞结构理论，篇章修辞关系定义在两个或多个基本篇章单元(EDU)之间。话语分割(Discourse Segmentation)的目标就是将篇章分割为基本篇章单元，从而实现后续的篇章分析任务。话语分割任务通常被形式化为序列标注任务或者单词级别的二分类任务，对每个单词位置输出预测其是否为一个基本篇章单元的边界。如图 5.6 所示，句子“The bank also says it will use its network to channel the investments”由三个基本篇章单元组成。本节将分别介绍两个经典的话语分割算法：基于词汇句法树的统计话语分割和基于循环神经网络的话语分割。

图 5.6 话语分割样例^[7]

5.2.1 基于词汇句法树的统计话语分割

SynDS^[7] 算法采用基于句法树的统计模型估计一句话中每个词作为分界点的概率。具体来说，给定句子 $s = \{w_1, w_2, \dots, w_n\}$ ，首先使用句法分析工具得到该句子的句法树 t ，随后对句子中的每个词 w_i ，使用最大似然估计的方法学习其作为分界点的概率 $P(b_i | w_i, t)$ ，其中 $b_i \in \{\text{boundary}(边界), \text{non-boundary}(非边界)\}$ 。由于句子间的分界点较易得到，SynDS 重点关注句子内部的分界，因此将句子间的分界设为 $P(\text{boundary} | w_n, t) = 1$ 。

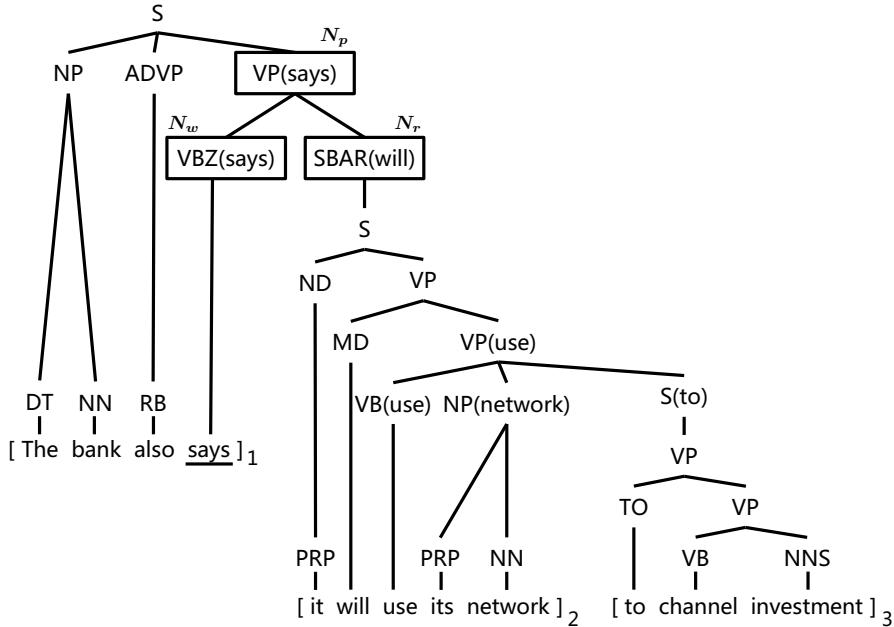
为了同时使用词汇及句法特征估计基本篇章单元分界，SynDS 算法使用文献 [8] 中提出词汇中心 (Lexical Head) 映射规则将词汇成分引入句法树。对于每个词 w 来说，SynDS 算法关注其包含一个右兄弟节点的最高父节点，使用其构建的特征决定当前词是否作为分界词。具体来说，将词 w 对应的带词汇信息的节点记为 N_w ，使用的特征包含 N_w 本身、其父节点 N_p 及其兄弟节点。例如，针对图5.7的例子，SynDS 在判断词“says”是否为分界词时，使用该词本身对应的节点 $N_w = VBZ(says)$ 、其父节点 $N_p = VP(says)$ 及其右兄弟节点 $N_r = SBAR(will)$ 作为特征。

为了训练分类模型，使用 RST-DT^[9] 语料的统计量估计每个词作为分界词的似然概率：

$$P(b | w, t) \simeq \frac{Cnt(N_p \rightarrow \dots N_w \uparrow N_r \dots)}{Cnt(N_p \rightarrow \dots N_w N_r \dots)} \quad (5.1)$$

其中，分子表示规则 $Cnt(N_p \rightarrow \dots N_w N_r \dots)$ 的在语料中出现且 w 为分界词的次数 (\uparrow 表示该处为分界)，分母表示该规则在语料中出现的总次数。

当通过统计模型获得每个词作为分界词的概率后，SynDS 算法对于给定一个句法树 t ，当 $P(\text{boundary} | w, t) > 0.5$ 时，选择 w 作为分界词（即在 w 后插入分界）。

图 5.7 基于词汇句法树的统计话语分割样例^[9]

5.2.2 基于循环神经网络的话语分割

话语分割任务还可以转换为序列标注问题^[10], 给定一个输入句子 $x = \{x_t\}_{t=1}^n$, 其输出 $y = \{y_t\}_{t=1}^n$ 中每个 y_t 表示第 t 个词是否为一个基本单元的开头, 如果是, 则 $y_t = 1$, 否则 $y_t = 0$ 。可以采用基于 BiLSTM-CRF 模型实现这一任务, 模型架构如图5.8所示。

将每个词 x_t 表示为一个向量 e_t , 然后使用一个双向 LSTM 网络 (Bi-LSTM) 建模序列中每个词的表示:

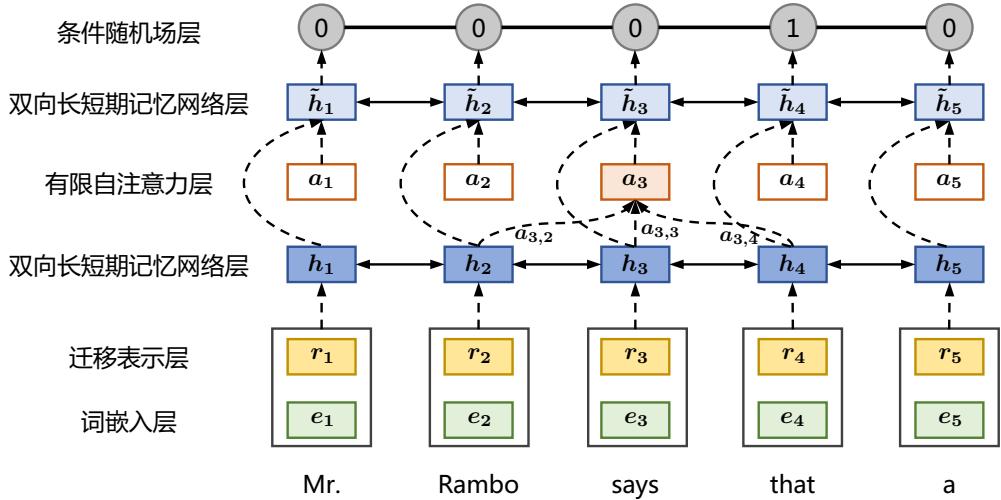
$$\mathbf{h}_t = \text{BiLSTM}(\mathbf{h}_{t-1}, e_t) \quad (5.2)$$

其中 $\mathbf{h}_t = [\mathbf{h}_t^f, \mathbf{h}_t^b]$ 为 t 位置正向 LSTM 及反向 LSTM 编码得到的隐向量表示的串联。

在获得每个词的隐向量表示后, 为了更好地利用序列信息, 使用条件随机场层进行输出解码。给定一个句子 x 的输出隐向量序列 $\mathbf{h} = \{\mathbf{h}_t\}_{t=1}^n$, 该句子预测为序列 y 的概率为:

$$P(y | \mathbf{h}; \mathbf{W}; \mathbf{b}) = \frac{\prod_{i=1}^n \psi_i(y_{i-1}, y_i, \mathbf{h})}{\sum_{y' \in \mathcal{Y}} \prod_{i=1}^n \psi_i(y'_{i-1}, y'_i, \mathbf{h})} \quad (5.3)$$

其中, \mathcal{Y} 表示所有可能的输出标签序列, $\phi_i(y'_{i-1}, y'_i, \mathbf{h}) = \exp(\mathbf{w}^T \mathbf{h}_i + b)$ 为势函数, \mathbf{w} 和 b 为和标签对 (y'_{i-1}, y'_i) 相关的权重和偏置。在训练时, 模型最大化输出正确标签序列的似然概率。在解

图 5.8 基于循环神经网络的话语分割^[10]

码时，模型计算产生最大似然概率的标签序列（关于条件随机场的具体介绍可以参见??章节）：

$$\mathbf{y}^* = \arg \max_{\mathbf{y}} P(\mathbf{y} | \mathbf{h}; \mathbf{W}; \mathbf{b}) \quad (5.4)$$

由于用于话语分割训练的语料通常较小，使用上述 BiLSTM 模型在该语料上训练难以取得理想地效果。因此，可以使用预训练词向量将更大语料上获取的知识迁移到话语分割任务上。例如使用 ELMo^[11] 词向量建模输入序列：

$$\mathbf{r}_t = \gamma^{LM} \sum_{l=0}^3 s_l^{LM} \mathbf{h}_{t,l}^{LM} \quad (5.5)$$

其中， \mathbf{s}^{LM} 为归一化权重，对 ELMo 词向量的三个组成部分进行加权， γ^{LM} 为整个 ELMo 词向量的权重。随后， \mathbf{r}_t 被拼接到词向量 \mathbf{e}_t 上作为模型的输入。

此外，由于许多 EDU 边界的预测需要长距离信息，而 LSTM 模型较难处理长距离依赖，还可以考虑使用自适应机制 (Restricted Self-attention) 加强句子的长距离依赖建模。但是并不是所依赖的距离越长越好，模型关注过长距离的信息可能会引入噪音，不利于预测。考虑 EDU 边界识别任务特性，即通常只需要使用相邻 EDU 的信息。因此，使用距离限制的自适应机制，只使用邻近的信息来预测。首先计算当前词 x_i 和一个窗口内的相邻词 x_j 之间的相似度：

$$s_{i,j} = \mathbf{W}_{attn}^T [\mathbf{h}_i, \mathbf{h}_j, \mathbf{h}_i \odot \mathbf{h}_j] \quad (5.6)$$

随后，每个词的注意力向量由相邻词加权获得：

$$\alpha_{i,j} = \frac{e^{s_{i,j}}}{\sum_{k=-K}^K e^{s_{i,i+k}}} \mathbf{a}_i = \sum_{j=-K}^K \alpha_{i,i+k} \mathbf{h}_{i+k} \quad (5.7)$$

其中， K 为使用的窗口大小。该注意力向量随后和 \mathbf{h} 一起被输入另一层 BiLSTM，并输出 $\tilde{\mathbf{h}}$ 作为 CRF 的输入：

$$\tilde{\mathbf{h}}_t = \text{BiLSTM}(\tilde{\mathbf{h}}_{t-1}, [\mathbf{h}_t, \mathbf{a}_t]) \quad (5.8)$$

最后，利用序列标注模型 CRF 给出输出。

5.3 篇章结构分析

篇章结构分析的目标是分析篇章单元之间存在的连贯关系，从而服务于下游任务。现有的篇章分析工作基于不同的篇章分析标注框架，主要可以分为两大类：基于词汇的浅层篇章分析及基于语义或意图关系的完整篇章分析。前者的代表性框架为^[12] 所提出的 Penn Discourse Treebank (PDTB) 标注框架；后者的代表性框架为基于修辞结构理论的 RST Discourse TreeBank (RST-DT) 标注框架^[9]。本节将介绍基于这两种代表性标注框架的篇章分析方法。

5.3.1 修辞结构篇章分析

RST-DT^[9] 是篇章分析中的代表性标注框架，其标注基于修辞结构理论，将一个完整的篇章标注成由基本篇章单元组成的层次树状结构。其中，树的节点关系由相邻篇章单元之间的关系构成。在标注时，一个完整的篇章首先被切分成不相交的基本篇章单元 (Elementary Discourse Units, EDU)，随后，基于修辞关系理论，相邻的基本篇章单元被连接并标注为 78 种修辞关系的一种。篇章整体最终被标注为层次化的树状结构。图 5.9 中展示了一个篇章的 RST-DT 标注样例。

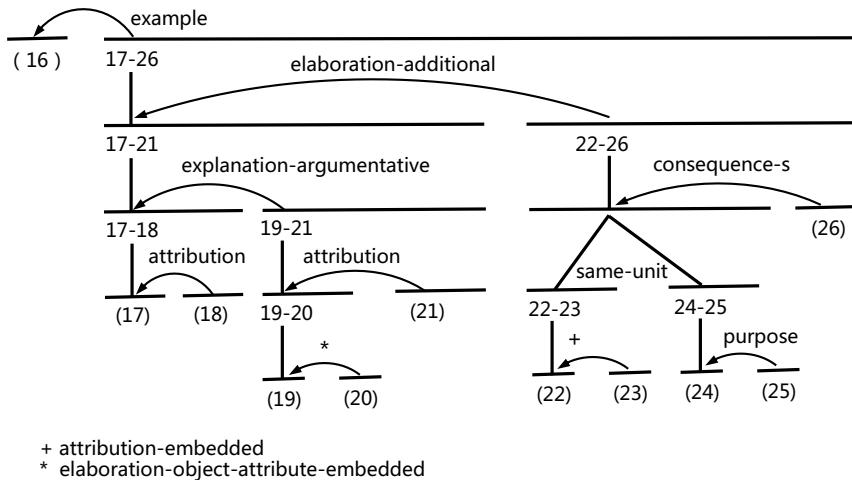
1. 基于 SVM 分类器的 RST 篇章分析

HILDA (HHigh-Level Discourse Analyzer)^[13] 是基于 SVM 分类器的 RST 篇章分析算法，将修辞结构树定义为二叉树结构，并定义了建立一个有效修辞结构树 (valid RS-tree) T 的两项规则：

- (1) T 的所有叶子结点均为 EDU (单个 EDU 也可以构成一个修辞结构树)。
- (2) T 的所有非叶子结点被标注为篇章关系集合中的一种关系 ($R_i \in \mathcal{R}$)。

基于这一修辞结构树的定义，HILDA 采用了基于贪心原则的流水线方法，使用两个支持向量机分类器对 EDU 之间是否存在关系以及存在何种关系分别进行分类。具体来说，HILDA 定义的两个分类器：

- 结构分类器 $\text{Struct}(l_i, l_j)$ ：用于判断篇章结构的二元分类器，即判断两个有效修辞结构树之间是否存在修辞关系，分类目标为 0 和 1，0 表示没有关系，1 表示有关系。
- 类型分类器 $\text{Label}(l_i, l_j)$ ：用于判断修辞关系类型及核类型的多元分类器，分类目标为篇章关

图 5.9 RST-DT 标注样例^[9]

系集合 $\mathcal{R} = \{R_1, \dots, R_n\}$, $R_i = <RR_i, Left_i, Right_i>$ 定义为由两个有效修辞结构树的核类型及其之间的修辞关系类型组成的三元组, 其中 $RR_i \in \{\text{ATTRIBUTION}, \text{CAUSE}, \dots\}$ 为 HILDA 所使用的修辞关系集合中的一种关系; $Left_i, Right_i \in \{\text{Nucleurs}, \text{Satellites}\}$ 为两个修辞结构树的核类型。

基于上述两个分类器, HILDA 的算法流程如算法5.1 所示。首先对输入文本, 创建一个包含所有 EDU 的列表, 其中 EDU 的排序按照从左到右的阅读顺序。当列表元素数目大于 1 时, 使用结构分类器计算列表中所有相邻单元的结构预测分数。基于贪心原则, 取出所有结构预测中分数最高的一组, 使用类型分类器预测其修辞关系类型及核类型, 并基于预测结果建立一个新的子树。得到新的子树后, 分别重新计算该子树与相邻单元之间的结构预测分数, 并将列表中对应单元替换为新的子树。重复上述过程直到列表元素数目等于 1, 则留在列表中的元素即为输出的篇章树。

为了能够更准确地对篇章结构及修辞关系进行分类, HILDA 构建了多种类型的特征作为 SVM 分类器的输入, 包括 N-gram 特征、句法结构特征、POS 特征等。Feng 等人在 HILDA 的基础上, 通过构建上下文等更丰富的语言特征进一步提升了性能^[14]。

2. 基于递归神经网络的 RST 篇章分析

Li 等提出了基于递归神经网络的 RST 篇章分析算法 RNN-RST^[15]。与 HILDA 算法相似, RNN-RST 也是通过训练两个分类器, 即结构分类器及修辞关系类型分类器构造修辞结构树, 但用于分类的特征则使用递归神经网络进行计算。算法的整体结构如图5.10所示。

具体来说, 对于每个给定句子 $S = \{w_1, \dots, w_{n_s}\}$, w_i 代表句子中的第 i 个词, n_s 代表句子中词的总数。每个词映射为一个词向量 $e_w \in \mathbb{R}^K$, 其中 K 为词向量的维度。对于给定句子, 计算它

代码 5.1: HILDA 算法

```

输入: EDU 列表  $E = \langle e_1, e_2, \dots \rangle$ 
输出: 篇章树  $FinalTree$ 

// 初始化
 $L \leftarrow E$  ;
foreach  $(l_i, l_{i+1})$  in  $L$  do
|    $Scores[i] \leftarrow Struct(l_i, l_{i+1})^a$ ; // 使用结构分类器计算列表中所有相邻单元的结构预
|   测分数 ;
end
// 解码过程
while  $|L| > 1$  do
|    $i \leftarrow \arg \max(Scores)$ ; // 取出结构预测分数最高的一组相邻单元 ;
|    $NewLabel \leftarrow Label(l_i, l_{i+1})$ ; // 预测修辞关系类别 ;
|    $NewSubTree \leftarrow CreateTree(l_i, l_{i+1}, NewLabel)^b$ ; // 建立新的子树 ;
|   // 更新结构预测列表
|    $Scores[i - 1] \leftarrow Struct(l_{i-1}, NewSubTree)$ ;
|    $Scores[i + 2] \leftarrow Struct(NewSubTree, l_{i+2})$ ;
|    $Delete(Scores[i])$ ;
|    $Delete(Scores[i + 1])$ ;
|    $L \leftarrow [l_0, \dots, l_{i-1}, NewSubTree, l_{i+2}, \dots]$ ; // 更新子树列表 ;
end
 $FinalTree \leftarrow l_0$  ;
return  $FinalTree$ 

```

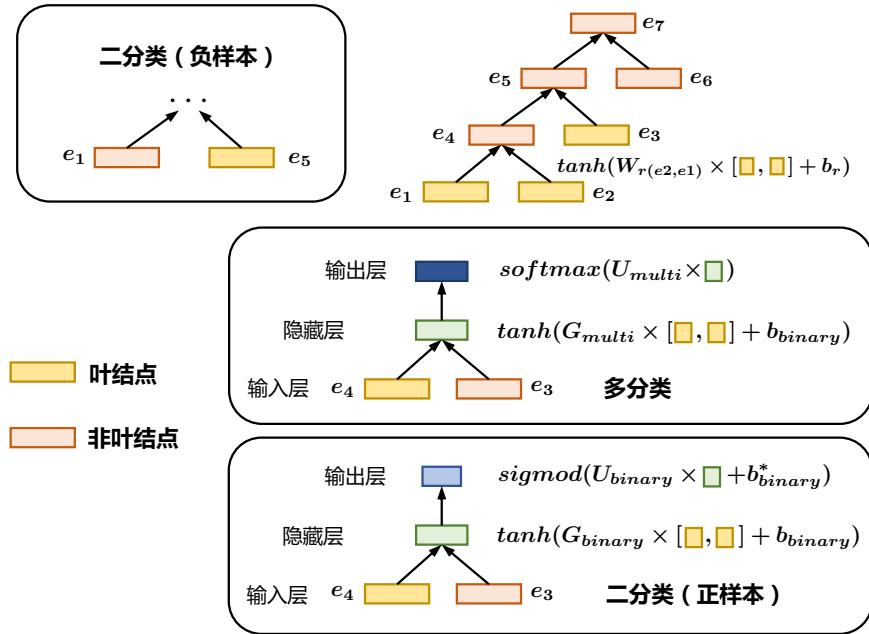
的特征向量 $\mathbf{h}_s \in \mathbb{R}^K$ 。

首先，使用句法分析工具对给定句子进行分析，分析得到的每个子句被视为一个基本篇章单元 EDU。基于得到的句法树，对于其中的每个父节点 p 及它的两个子节点 c_1 和 c_2 （分别对应向量表示 \mathbf{h}_{c_1} 和 \mathbf{h}_{c_2} ），该父节点的表示计算如下：

$$\mathbf{h}_p = f(\mathbf{W} \cdot [\mathbf{h}_{c_1}, \mathbf{h}_{c_2}] + \mathbf{b}) \quad (5.9)$$

其中 $[\mathbf{h}_{c_1}, \mathbf{h}_{c_2}]$ 为子节点向量表示 \mathbf{h}_{c_1} 和 \mathbf{h}_{c_2} 的拼接， \mathbf{W} 为一个 $K \times 2K$ 的矩阵， \mathbf{b} 为 $1 \times K$ 的偏移向量。 $f(\cdot)$ 为 \tanh 激活函数。对于整个句法树，递归神经网络由下至上递归计算每个父节点的表示，直到获得该句的根结点的表示，并将该表示作为该句子的向量表示 \mathbf{h}_s 。

RNN-RST 同样使用结构分类器和修辞关系类型分类器分别对 EDU 之间的结构和关系进行分类。结构分类器使用单层卷积神经网络编码，并使用单层线性分类器投影至 $[0,1]$ 输出空间进行

图 5.10 基于递归神经网络的 RST 篇章分析算法神经网络结构图^[15]

分类:

$$\begin{aligned}
 L_{(e_i, e_j)}^{binary} &= f(\mathbf{G}_{binary} * [\mathbf{h}_{e_i}, \mathbf{h}_{e_j}] + \mathbf{b}_{binary}) \\
 P[t_{binary}(e_i, e_j) = 1] &= g(\mathbf{U}_{binary} \cdot L_{(e_i, e_j)}^{binary} + b_{binary}^*)
 \end{aligned} \tag{5.10}$$

其中, \mathbf{G}_{binary} 是一个 $N_{binary} \times 2K$ 的卷积矩阵, \mathbf{b}_{binary} 是偏移向量, $f(\cdot)$ 为 \tanh 激活函数; \mathbf{U}_{binary} 是一个 $N_{binary} \times 1$ 的向量, b_{binary}^* 表示偏移值, $g(\cdot)$ 为 sigmoid 激活函数。 $t_{binary}(e_i, e_j) = 1$ 代表两个 EDUE e_i 和 e_j 之间存在依存关系。

当 $t_{binary}(e_i, e_j)$ 的预测值为 1 时, 接着使用一个多元分类器预测其修辞关系类型, 预测的关系类型表示为 $r(e_i, e_j)$ 。修辞关系类型分类器的结构和结构分类器类似, 但使用 Softmax 激活函数进行输出:

$$\begin{aligned}
 L_{(e_i, e_j)}^{multi} &= f(\mathbf{G}_{multi} * [\mathbf{h}_{e_i}, \mathbf{h}_{e_j}] + \mathbf{b}_{multi}) \\
 S_{(e_i, e_j)} &= \mathbf{U}_{multi} \cdot L_{(e_i, e_j)^{multi}} \\
 P_{(e_i, e_j)}(i) &= \frac{\exp(S_{(e_i, e_j)}(i))}{\sum_k \exp(S_{(e_i, e_j)})(k)}
 \end{aligned} \tag{5.11}$$

其中, \mathbf{G}_{multi} 是一个 $N_{multi} \times 2K$ 的矩阵, \mathbf{b}_{multi} 是偏移向量, $f(\cdot)$ 为 \tanh 激活函数。 \mathbf{U}_{multi} 是一个 $N_r \times 2K$ 的矩阵, $P_{(e_1, e_2)}$ 中的第 i 个元素代表 e_i 和 e_j 之间存在第 i 种关系的概率。需要注意的是, 二元分类器及多元分类器在训练时为分别训练。

上述的分类计算基于每个节点的表示进行。然而, 公式5.10只能获得每个句子即叶子节点的表示, 而无法获得非叶子节点的表示。对于非叶子节点, 即由 EDU 构成的子树, 该方法使用递归神经网络进一步由下至上计算其父节点。具体地, 对于给定子节点表示 h_{e_i} 和 h_{e_j} 及其标注类型 $r(e_i, e_j)$, 其父节点表示计算如下:

$$\mathbf{h}_p = f(\mathbf{W}_{r(e_i, e_j)} \cdot [\mathbf{h}_{e_i}, \mathbf{h}_{e_j}] + \mathbf{b}_{r(e_i, e_j)}) \quad (5.12)$$

其中, $\mathbf{W}_{r(e_i, e_j)}$ 为 $r(e_i, e_j)$ 所对应关系类型的 $K \times 2K$ 参数矩阵, $\mathbf{b}_{r(e_i, e_j)}$ 为该关系类型对应的偏移向量, $f(\cdot)$ 为 \tanh 激活函数。

利用标注语料集合, 可以分别构造上述两个分类器的训练数据, 并利用交叉熵损失函数进行模型参数训练。在训练完成后可以采用类似用于句法分析的 CKY 动态规划方法, 对于给定的篇章进行修辞结构树构建。对于由 n 个 EDU 组成的篇章, 可以构建 $N_r \times n \times n$ 组成的动态规划表 Pr , N_r 表示关系类型数量, Pr 表中每个单元格 $Pr[r, i, j]$ 表示从片段从第 i 个 EDU 到第 j 个 EDU 中具有关系 r 的概率, 其计算过程如下:

$$\begin{aligned} Pr[r, i, j] &= \max_{r_1, r_2, k} Pr[r_1, i, k] \cdot Pr[r_2, k, j] \\ &\quad \times P(t_{binary}(e_{[i, k]}, e_{[k, j]}) = 1) \\ &\quad \times P(r(e_{[i, k]}, e_{[k, j]}) = 1) \end{aligned} \quad (5.13)$$

5.3.2 浅层篇章分析

Penn Discourse Treebank (PDTB) 是基于词汇化树型连接语法 (Discourse Lexical Tree Adjunct Grammar, D-LTAG) 理论^[16] 构建的篇章分析标注框架, 是篇章分析中的另一代表性框架。PDTB 以篇章内相邻或者跨度在一定范围内的片段, 以连接词为核心, 对片段间关系进行标注。相较于修辞结构理论将整个篇章构建为树结构而言, PDTB 则针对两个片段之间的关系, 因此也可以称为浅层篇章分析。每个篇章关系由两个论据 (Argument) 及其之间的关系组成, 两个论据分别标注为 Arg1 和 Arg2。在相邻句子构成的关系中, Arg1 和 Arg2 则反映论据之间的线性顺序, 其中 Arg1 在 Arg2 之前。根据连接词是否显式存在, PDTB 标注的关系可分为显式篇章关系和隐式篇章关系两类。

显式篇章关系 (Explicit Discourse Relation) 由显式连接词定义, 通过显式连接词连接 Arg1 和 Arg2。在显式关系中, Arg2 一般为句法上关联的论据, Arg1 则为另一个论据, 显式连接词由三种语法连接词产生:

- 从属连词，如 because, when 等；
- 并列连词，如 and, or 等；
- 语篇副词，如 for example, instead 等。

除此之外，还有一些带修饰或联合形式的连接词，如 only because, if and when 等；及小部分并列连接词，如 either..or, on the one hand..on the other hand 等。以下是一些基于显式关系定义的标注样例^[12]（下划线标注显式连接词，斜体标注 Arg1，粗体标注 Arg2）：

- (1) *Third-quarter sales in Europe were exceptionally strong, boosted by promotional programs and new products –although weaker foreign currencies reduced the company’s earnings.*
- (2) *Most oil companies, when they set exploration and production budgets for this year, forecast revenue of \$15 for each barrel of crude produced.*

隐式篇章关系（Implicit Discourse Relation）则是除显式篇章关系以外，需要靠读者通过推断判断的篇章关系。例如下面的例句：

- (1) But a few funds have taken other defensive steps. *Some have raised their cash positions to record levels. Implicit = BECAUSE High cash positions help buffer a fund when the market falls.*

虽然没有显式连接词，但读者能够通过论据之间的语义判断出其之间表达的因果关系。在 PDTB 中，这样的隐式关系通常通过标注者插入一个连接词进行标注（例如上面例子中的 BECAUSE 被插入以表示因果关系）。而当隐式关系无法使用一个隐式连接词进行标注时，则构成三种特殊的隐式关系：

- **AltLex** 表示语篇关系已经由非连接词的词汇表达，额外插入连接词会构成冗余的情况；
- **EntRel** 表示句子之间只存在基于实体的连贯关系的情况。
- **NoRel** 表示句子之间不存在任何篇章关系或基于实体的连贯关系的情况。

下面三个例子分别为 **AltLex**、**EntRel**、**NoRel** 的标注样例：

- (1) *Ms. Bartlett’s previous work, which earned her an international reputation in the non-horticultural art world, often took gardens as its nominal subject. Implicit = AltLex **Mayhap this metaphorical connection made the BPC Fine Arts Committee think she had a literal green thumb.***
- (2) *Hale Milgrim, 41 years old, senior vice president, marketing at Elecktra Entertainment Inc., was named president of Capitol Records Inc., a unit of this entertainment concern. Implicit = EntRel **Mr. Milgrim succeeds David Berman, who resigned last month.***
- (3) *Jacobs is an international engineering and construction concern. Implicit = NoRel **Total capital investment at the site could be as much as \$400 million, according to Intel.***

由于一个连接词在不同的篇章中可能表达不同的语义关系，PDTB 中为显式关系、隐式关系和 AltLex 关系提供了三级语义标注（Sense Tag）：CLASS, TYPE, SUBTYPE。其中，最高层标注（CLASS）包含四个主要语义类别：TEMPORAL, CONTINGENCY, COMPARISON, EXPANSION。对于每个语义类别，使用 TYPE 进一步标注其语义。第三级语义标签 SUBTYPE 则具体化每个论

据的语义贡献。图5.11展示了PDTB的三级语义标签。

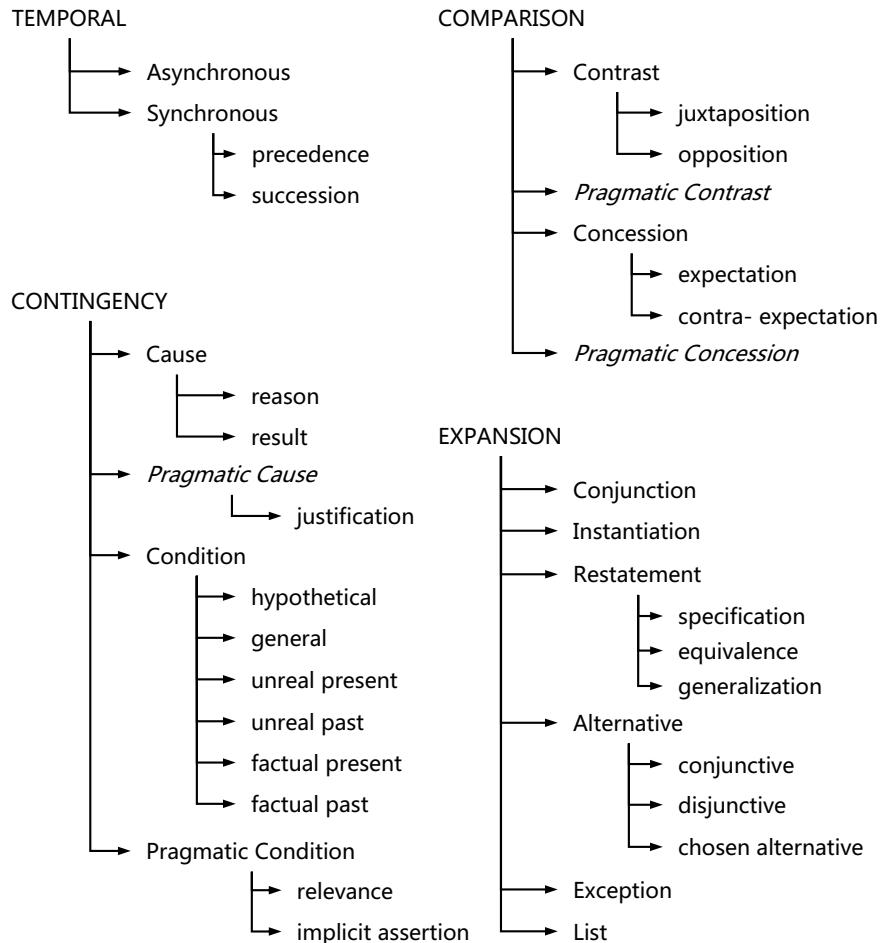


图 5.11 PDTB 标注的三级语义标签

基于上述 PDTB 标注的篇章分析工作一般可以划分为显式篇章分析（Explicit Discourse Parsing）和隐式篇章分析（Implicit Discourse Parsing）两类。其中，显式篇章分析关注篇章的显式关系识别，包括显式连接词检测、论据标注等子任务；隐式篇章分析则更关注给定句子对之间的隐式关系分类。本节将分别介绍显式篇章分析和隐式篇章分析的代表性工作。

1. 基于句法特征构建的显式篇章分析

由于部分显式连接词在不同语境下具有不同语义，显式篇章分析的重点在于对显式连接词进行消歧，并将每一连接词分类为 PDTB 标注的四个一级语义类别 (TEMPORAL、CONTINGENCY、COMPARISON、EXPANSION)。例如，在下述两个例子中：

(1) Guangzhou has a wide water area with many rivers and water systems since it is located in the water-rich area of southern China.

(2) She has been living in Beijing since she graduated from Fudan University.

显式连接词“Since”在句子(1)中表示因果关系，为 CONTINGENCY 语义类别；在句子(2)中则为 TEMPORAL 语义类别。显式篇章分析关注的主要问题即为将连接篇章中话语的显式连接词正确划分为其所属的语义类别，这一任务通常以文本分类的方式实现。

Pitler 等人^[17]使用最大熵分类器，通过利用句法特征对显式篇章关系进行分类。其基于标准 Penn Treebank 句法分析标注^[18]，构建了多种句法特征对显式连接词的语义进行消歧，所构建的句法特征包括：

- 自身类别 (Self Category)：子树包含且仅包含该显式连接词的最高父节点。对于单个单词构成的显式连接词，其特征为该词自身的 POS 标注；对于多个单词构成的显式连接词则不然。例如，*in addition* 的成分句法标注为 (PP (IN In) (NP (NN addition))), 其自身类别则为 PP (Prepositional Phrase)。
- 父节点类别 (Parent Category)：自身类别的最近父节点的类别。
- 左兄弟节点类别 (Left Sibling Category)：离自身类别最近的左兄弟节点类别。如果左兄弟节点不存在，则其特征为“None”。
- 右兄弟节点类别 (Right Sibling Category)：离自身类别最近的右兄弟节点类别。文献 [17] 认为，由于英语是右分支结构，句子的依赖一般出现在其头部之后，因此右兄弟节点类别一般包含该显式连接词的依赖，从而显得尤为重要。对于表达篇章关系的显式连接词，其依赖一般为一个从句。例如，句子“*After I went to the store, I went home*”可以通过右兄弟节点类别显示其表达的篇章关系，从而和句子“*After May, I will go on vacation*”区分开。除了右兄弟节点类别以外，作者还增加了两项特征以进一步利用右兄弟节点的信息，提升消歧效果：包含一个 VP 的右兄弟节点 (Right Sibling Contains a VP) 和包含一个 Trace 的右兄弟节点 (Right Sibling Contains a Trace)。

Pitler 等人^[17]的实验结果表明，通过构建句法特征，对显式连接词进行分类能够达到 94.15% 的准确率。因此，后续 PDTB 篇章分析的工作更多关注隐式篇章分析。

2. 基于循环神经网络语言模型的隐式篇章分析

由于传统机器学习方法在显式篇章分析任务上已经能够达到较高的准确率^[19]，后续基于 PDTB 的篇章分析工作更多地关注隐式篇章分析，相关的工作包括基于前馈网络^[20]、基于浅层卷积神经

网络^[21]、基于循环神经网络语言模型^[22] 及基于预训练语言模型^[23] 的隐式篇章分析等。本节中将介绍基于基于循环神经网络语言模型（RNN 语言模型）的隐式篇章分析算法 DRLM^[22]。

DRLM^[22] 使用包含隐变量的循环神经网络语言模型建模隐式篇章分析算法，整个过程建模为两阶段生成过程。首先，句子 $t - 1$ 和句子 t 之间的隐式篇章关系 z_t 由句子 $t - 1$ 的信息建模。在此基础上，句子 x_t 根据句子 x_{t-1} 和 z_t 生成。DRLM 的网络结构如图5.12所示。整个模型被

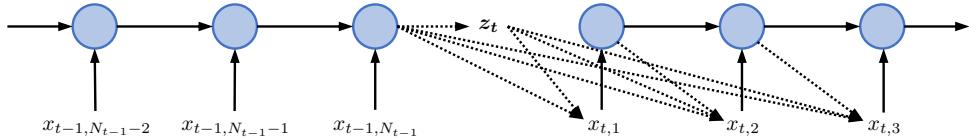


图 5.12 DRLM 算法包含隐变量结构图^[22]

给定输入句子 $\mathbf{x}_t = \{x_{t,n}\}_{n \in \{1 \dots N_t\}}$ ，其中 t 表示该句为篇章中的第 t 个句子， N_t 为句子 t 的长度。基于链式法则，RNN 语言模型将该句出现的概率转化为每个词出现的条件概率的乘积：

$$p(\mathbf{x}_t) = \prod_{n=1}^{N_t} p(x_{t,n} | \mathbf{x}_{t,< n}) \quad (5.14)$$

在每个时刻 n ，RNN 语言模型的输入为包含所有历史信息的上一刻输出隐变量 $\mathbf{h}_{t,n-1}$ 和当前词的词向量 $\mathbf{X}_{x_{t,n}}$ ，其预测下一个词的条件概率为：

$$\begin{aligned} \mathbf{h}_{t,n} &= f(\mathbf{X}_{x_{t,n}}, \mathbf{h}_{t,n-1}) \\ p(x_{t,n} | \mathbf{x}_{t,< n}) &= \text{softmax}(\mathbf{W}_o \mathbf{h}_{t,n-1} + \mathbf{b}_o) \end{aligned} \quad (5.15)$$

其中， $\mathbf{W}_o \in \mathbb{R}^{V \times K}$ 为输出层权重， $\mathbf{b}_o \in \mathbb{R}^V$ 为输出层偏移量。

由于篇章分析需要对包含多句话的长文本进行语言模型建模，而 RNN 语言模型难以处理长距离依赖关系，DRLM 使用了基于文档的语言模型^[24]。具体来说，文档中第 t 个句子的第 n 步输出的条件概率为：

$$p(x_{t,n} | \mathbf{x}_{t,< n}, \mathbf{x}_{< t}) = \text{softmax}(\mathbf{W}_o \mathbf{h}_{t,n-1} + \mathbf{W}_c \mathbf{c}_{t-1} + \mathbf{b}_o) \quad (5.16)$$

其中， \mathbf{c}_{t-1} 为句子 $t - 1$ 的上下文信息，此处设为上一个句子的最后一步输出的隐向量。

基于上述 RNN 语言模型，DRLM 将篇章关系 z_t 作为隐变量引入，并设计了两步语言模型生成过程，如图5.12所示。第一步，使用句子 $t - 1$ 的上下文信息 \mathbf{c}_{t-1} 生成句子 $t - 1$ 和句子 t 之间

的篇章关系：

$$p(z_t | \mathbf{x}_{t-1}) = \text{Softmax}(\mathbf{U}\mathbf{c}_{t-1} + \mathbf{b}) \quad (5.17)$$

其中， z_t 是一个表示句子间篇章关系的随机变量。

第二步，基于句子 \mathbf{x}_{t-1} 及上一步生成的篇章关系 z_t ，生成句子 \mathbf{x}_t ：

$$p(\mathbf{x}_t | z_t, \mathbf{x}_{t-1}) = \prod_n^{N_t} p(x_{t,n} | \mathbf{x}_{t,< n}, \mathbf{x}_{t-1}, z_t) \quad (5.18)$$

其中，引入篇章关系隐向量的输出条件概率计算为：

$$p(x_{t,n} | \mathbf{x}_{t,< n}, \mathbf{x}_{t-1}, z_t) = g(\mathbf{W}_o^{(z_t)} \mathbf{h}_{t,n} + \mathbf{W}_c^{(z_t)} \mathbf{c}_{t-1} + \mathbf{b}_o^{(z_t)}) \quad (5.19)$$

$\mathbf{W}_o^{(z_t)} \mathbf{h}_{t,n}$ 、 $\mathbf{W}_c^{(z_t)} \mathbf{c}_{t-1}$ 及 $\mathbf{b}_o^{(z_t)}$ 为由篇章关系决定的参数，不同的篇章关系通过不同的权重关注特征空间中的不同部分特征。

最后，文本及篇章关系的联合概率为：

$$p(\mathbf{x}_{1:T}, z_{1:T}) = \prod_t^T p(\mathbf{x}_t | z_t, \mathbf{x}_{t-1}) \times p(\mathbf{x}_t | z_t, \mathbf{x}_{t-1}) \quad (5.20)$$

在训练阶段，DRLM 可以使用两种目标函数进行训练：联合似然目标函数和条件目标函数。其中，联合似然目标函数的损失函数计算为：

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_t^T \log p(z_t | \mathbf{x}_{t-1}) + \sum_n^{N_t} \log p(x_{t,n} | \mathbf{x}_{t,< n}, \mathbf{x}_{t-1}, z_t) \quad (5.21)$$

其中 $\boldsymbol{\theta}$ 表示模型参数。使用上述联合似然目标函数能够同时优化模型的篇章关系预测能力及语言模型能力，可以视为一种多任务学习。然而，在实际实现时，由于词的数量比句子的数量更多，使用这一目标对模型语言模型能力的优化占主导地位。

条件目标函数的损失函数计算为：

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_t^T \log p(z_t | \mathbf{x}_{t-1}) + \log p(\mathbf{x}_t | z_t, \mathbf{x}_{t-1}) - \log \sum_{z'} p(z' | \mathbf{x}_{t-1}) \times p(\mathbf{x}_t | z', \mathbf{x}_{t-1}) \quad (5.22)$$

该式的前两项与公式 5.21 一致，但第三项计算了所有可能的 z' 生成句子 \mathbf{x}_t 的损失。这项损失的加入使得目标函数倾向于优化篇章关系相关的条件似然损失，而将语言模型任务视为一个辅助任务。

在推断时，通过贝叶斯公式计算句子 $t - 1$ 和句子 t 之间为关系 z_t 的条件概率：

$$p(z_t | \mathbf{x}_t, \mathbf{x}_{t-1}) = \frac{p(\mathbf{x}_t | z_t, \mathbf{x}_{t-1}) \times p(z_t | \mathbf{x}_{t-1})}{\sum_{z'} p(\mathbf{x}_t | z', \mathbf{x}_{t-1}) \times p(z' | \mathbf{x}_{t-1})} \quad (5.23)$$

其中的每项概率可以根据公式5.17和公式5.18进行计算。

5.4 指代消解

在本章第 5.1 节篇章衔接的概要介绍中，我们提到了篇章中的指代现象。该现象是体现篇章衔接性的重要组成部分。在本节中，我们将介绍篇章分析的另一重要子任务：指代消解（Coreference Resolution）。指代消解旨在将同一实体（entity）在篇章中出现的不同表述（mention）划分到同一等价类（或称表述集群）中。其中，实体指某一客观存在的事物；表述则为指代某一实体的表述。指代消解任务通常关注两种指代类型：共指（coreference）和回指（anaphora）。共指表示两个表述指向真实世界中的同一实体。

例如：上海的卖腌腊的店铺里也卖咸鸭蛋，必用纸条特别标明：“高邮咸蛋”。

（《端午的鸭蛋》）

上例中，“咸鸭蛋”和“咸蛋”指代真实世界中的统一实体，因此为共指关系。

回指表示当前表述指向上述出现的另一表述，通常将指代上文的表述称为照应词（Anaphor），将照应词指代的上文表述称为先行词（Antecedent）。

例如：我围抱着火炉，烤热漫长一生的一个时刻。我知道这一时刻之外，我其余的岁月，我的亲人们的岁月，远在屋外的大雪中，被寒风吹彻。

（《寒风吹彻》）

上例中，“这一时刻”指代上文的“一个时刻”，为回指关系。其中“这一时刻”为照应词，“一个时刻”为先行词。

指代消解任务将语篇中所有表示同一实体的指代分配到同一等价类中，并给出每一语篇中的所有等价类。

例如：其间有一个十一二岁的少年 [1]，项带银圈，手捏一柄钢叉，向一匹猹 [2] 用力地刺去。那猹 [2] 却将身一扭，反从他 [1] 的胯下逃走了。

（《少年闰土》）

上例中，“一个十一二岁的少年”和“他”指代同一实体，属于同一等价类；“一匹猹”和“那猹”指代同一实体，属于同一等价类。指代消解任务的目标是发现文中的等价类 [1] 和 [2]。

指代消解任务一般可分为两个步骤：表述发现（Mention Detection）和指代消解（Coreference Resolution）。其中，表述发现旨在找出句子中所有可能存在指代关系的名词表述，一般包含人称代词（“你”、“我”、“他”等）、命名实体（人名、地名等）及一些名词短语（“那只猫”、“右边的女士”）等。表述发现的方法一般更注重将所有的表述找出，即更注重提升召回率，并在之后对无关

的表述进行过滤。指代消解则是这一任务的核心，也是最具挑战的步骤，其旨在对表述同一实体的表述进行聚类。

指代消解的方法主要有基于表述对（mention pair）、基于表述排序（mention ranking）和基于实体等三种方法。其中，基于表述对的方法使用一个二分类分类器对每一对表述是否为指代关系进行打分。基于表述排序的方法针对给定指代，通过计算其与每一表述组成的表述对的分数，对相关表述进行排序，以确定其指代关系。基于聚类的方法对文本中所有表述进行聚类，每个类被认为指代同一实体名词。

5.4.1 基于表述对的指代消解

基于表述对的指代消解算法是指代消解的一类经典算法。此类算法利用二分类器，分别对每个表述与其所有先行词所构成的表述对是否构成指代关系进行分类。如图5.13所示，为对下述例子进行基于表述对指代消解的过程示意：

长妈妈，已经说过，是一个一向带领着我的女工，说得阔气一点，就是我的保姆。我的母亲和许多别的人都这样称呼她，似乎略带些客气的意思。

（《阿长与山海经》）

对于所选表述“她”，基于表述对的指代消解算法分别计算“她”和其所有先行词构成的表述对为指代关系的分数。在这一例子中，“长妈妈”和“她”为正确的指代关系，因此指代消解算法目标是对这一表述对输出较高的分数；“我的母亲”这一先行词与“她”不为指代关系，则指代消解算法将对这一表述对输出较低的分数。

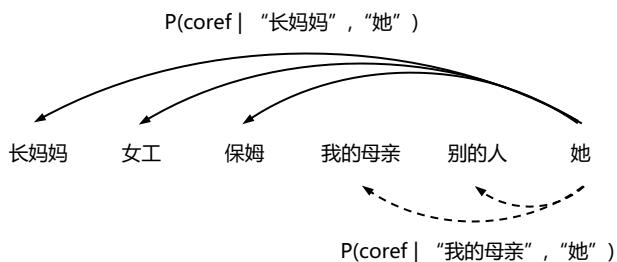


图 5.13 基于表述对的指代消解示例

在训练用于表述对分类的二分类器时，尽管训练语料中提供了表述之间的所有等价类标注，但由于相同指代的表述之间并没有直接的指代关系，将属于同一等价类的所有表述都视为正类训练可能不能取得很好的效果^[25]。因此，对于一个表述 m ，经典的策略是选择和当前表述存在指代关系的距离最近的先行词 a 与 m 构成的表述对 (a, m) 作为正样本，而选取所有和 m 不属于同一等价类的先行词 b 构成的表述对 (b, m) 作为负样本^[26]。

基于一个训练得到的用于表述对分类的二分类器，对每个测试文本的指代消解推理可以视为

一个构造消解图（coreference graph）的过程（也可视为聚类过程）：每个表述为图中的一个节点，当分类器预测一个对表述之间为指代关系时，则为这两个节点之间添加一条有向边。由此种方式构成的图，能够表示每两个表述之间是否互为指代；同时通过所有连接构成的传递闭包（transitive closure）我们能够找出所有等价类^[25]。为了实现这一目标，基于表述对的指代消解算法通常使用分类器为每个表述选择一个与其为指代关系的先行词，并将该表述与该先行词连接。对于该先行词的选择策略，基于最近原则（**Closest-First**）的算法^[26]从后向前依次计算所有先行词与该表述构成的表述对为指代关系的分数，并选择第一个大于某一阈值的先行词；基于最优原则（**Best-First**）的算法^[27]则计算所有先行词与该表述构成的表述对为指代关系的分数，并选出分数最高的先行词进行连接。

基于上述训练及推断框架，基于表述对的指代消解系统通常使用不同的模型作为判断表述对指代关系的二分类器。早期方法使用手工构造特征训练分类器^[25]，近年来基于深度神经网络的方法则使用神经网络学习用于分类的特征^[28]。

1. 基于特征工程的表述对指代解

Bengtson 等人构造了基于多类特征及感知器分类的表述对指代消解系统 Feature-pair^[25]。其构造的特征主要包括两个方面：表述特征及表述对特征。其中，表述特征包括表述类型特征，例如其是否为专有名词（proper noun）、普通名词（common noun）或代词（pronoun）等；表述对特征包括表述对的字符串关系特征（如一个字符串是否为另一个的子串）、语义相符性特征（例如性别、数字是否相符）、相对位置特征、实体类别特征等。具体特征描述如下：

- 表述类型特征：表述所属的类型，为专有名词（proper noun）、普通名词（common noun）或代词（pronoun）。
- 字符串关系特征：两个字符串之间是否存在一些共有特征，例如一个字符串是另一个字符串的子串等。
- 语义特征：包括两个名词之间的性别是否相符、数字是否相符；两个名词是否为近义词、反义词、或上位词等。
- 相对位置特征：两个表述之间的位置关系，例如将距离转化为二元特征 ($[distance \leq i]$, i 包括所有间隔值)、两个表述是否属于同一个句子等。
- 可学习特征：基于可学习分类器得到的特征，例如使用分类器判断两个由同一修饰语修饰的表述，其修饰语之间是否存在指代关系。
- 修饰语对齐特征：两个上位词相同的修饰语之间存在的关系，例如是否为子串、近义词、反义词等。
- 记忆特征：选取一些常常构成指代关系的表述对构造特征（例如“the queen”和“Elizabeth II”），供模型记忆学习。
- 预测实体类别特征：基于模版匹配预测实体所属的实体类别（人名、地名、机构名等），并基于预测类别构造两个表述之间实体类别是否匹配或是否相交等特征。

更具体特征分类可参考文献 [25]。

基于上述构造的特征, Feature-pair 训练一个感知器对每一指代对是否构成指代关系进行分类。训练时, Feature-pair 基于最近原则, 选择当前表述 m 与其所指代的距离最近的先行词 a 所构成的表述对 (a, m) 作为正样本; 选取在 m 之前所有和 m 不属于同一等价类的表述作为负样本。测试时, Feature-pair 基于最优原则, 每次选择与 m 构成分数最高的表述对的先行词:

$$a = \arg \max_{b \in B_m} (\text{PC}(b, m)) \quad (5.24)$$

其中 $\text{PC}(\cdot)$ 为表述对分数计算函数, 当 $\text{PC}(a, m)$ 大于某一预定义阈值时, 则将 a 和 m 链接为同一等价类。

2. 基于神经网络的表述对指代消解

Clark 等人构造了基于前馈神经网络的表述对指代消解分类器 Feedforward-pair^[28], 其结构如图5.14所示。其中, 输入层将输入词映射到输入特征空间, 其输入特征由表述及表述相关词的词嵌入向量及一些其他特征构成。表述相关词包括表述的依赖词、句法树中的父节点、表述的第一个词、表述的第二个词、表述之前的两个词及之后的两个词等等; 其他特征包括表述的类型特征、位置特征、文档类型特征等。

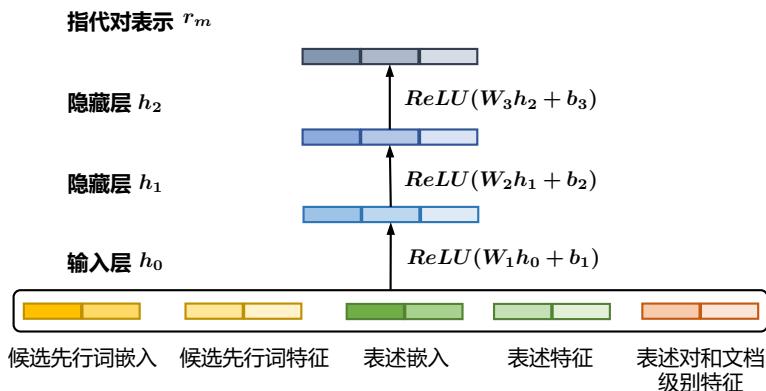


图 5.14 基于神经网络的表述对指代消解^[28]

基于构建的输入特征, 使用前馈网络和 ReLU 激活层构造三层前馈神经网络:

$$\mathbf{h}_i(a, m) = \max(0, \mathbf{W}_i \mathbf{h}_{i-1}(a, m) + \mathbf{b}_i) \quad (5.25)$$

其中, $\mathbf{h}_i(a, m)$ 为输入表述对 (a, m) 的第 i 层输出特征。 \mathbf{W}_i 为第 i 层的权重矩阵, \mathbf{b}_i 为第 i 层的

偏置向量。最后，模型得到第三层的输出特征 $\mathbf{h}_3(a, m)$ 用于分类。

5.4.2 基于表述排序的指代消解

基于表述对的指代消解基于二分类器分别对每个先行词和当前指代构成的表述对计算分数。然而，这种做法对于不同先行词的分数计算是相互独立的，只能判断每个先行词相对当前指代的合理程度，而无法直接通过比较判断哪个先行词是最正确的被指代词^[29]。为了解决这一问题，研究人员提出了基于表述排序的指代消解算法，如图5.15所示。其基本思路是使用一个多分类器，基于多个先行词候选计算出分数最高的先行词。

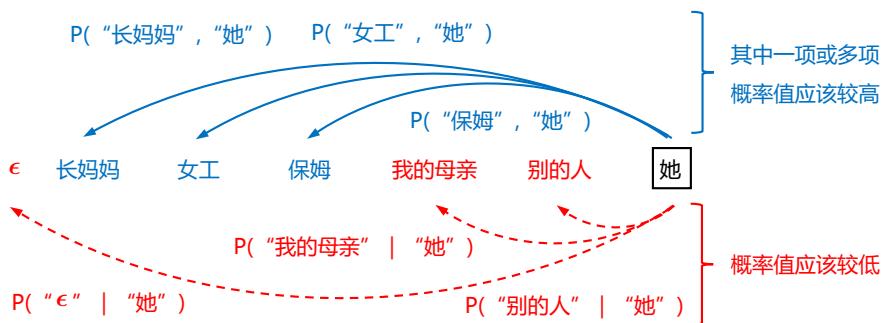


图 5.15 基于表述排序的指代消解示例

1. 基于特征工程和最大熵分类器的表述排序指代消解

Denis 等人提出了基于表述排序的指代消解算法 RK^[29]，该方法将指代消解任务从基于指代对二分类器的多步推理（先分别计算各指代对的分数，再基于某种策略选出最高分的先行词），转化为同时计算并比较所有先行词候选的单步推理过程。具体来说，对每个先行词候选 α_i ，模型计算其为当前指代词 π 的被指代先行词的条件概率 $P_r(\alpha_i | \pi)$ ，从而对于每个指代词 π ，通过比较多个候选先行词的条件概率即可以选出最可能和 π 构成指代关系的先行词：

$$P_r(\alpha_i | \pi) = \frac{\exp(\sum_{i=1}^n \lambda_i f_i(\pi, \alpha_i))}{\sum_k \exp(\sum_{i=1}^n \lambda_i f_i(\pi, \alpha_k))} \quad (5.26)$$

RK 算法通过最大熵分类器建模这一条件概率，其使用的特征包含三个类别：(1) 照应词特征，即描述待分类表述的特征，包括其代词类型特征、大小写特征等；(2) 候选先行词特征，即描述候选先行词的特征，包括其词性特征、其左右相关词的词性特征等；(3) 关系特征，即描述两个表述之间关系的特征，包括两个词之间的距离、两个词的语义相符性特征等。其使用的部分特征可见表5.1。

在训练时，RK 算法同样需要根据给定标注数据，为每个表述选取正负样本进行训练。对于任

表 5.1 RK 算法使用的部分特征^[29]

代词特征	
PERS_PRO	如果 π 是人称代词则为 T, 否则为 F
POSS_PRO	如果 π 是所有格代名词则为 T, 否则为 F
THIRD_PERS_PRO	如果 π 是第三人称代词则为 T, 否则为 F
SPEECH_PRO	T 如果 π 是第一或第二人称代词则为 T, 否则为 F
PRO_FORM	T 如果 π 是小写字母组成的代词则为 T, 否则为 F
候选先行词特征	
ANTE_WD_LEN	α 中单词的数量
PRON_ANTE	如果 α 是代词则为 T, 否则为 F
PN_ANTE	如果 α 是专有名词则为 T, 否则为 F
INDEF_ANTE	如果 α 是无定名词短语 (indefinite NP) 则为 T, 否则为 F
DEF_ANTE	如果 α 是有定名词短语 (definite NP) 则为 T, 否则为 F
关系特征	
S_DIST	π 和 α 之间句子数量的分桶值 (Binned values)
NP_DIST	π 和 α 之间表述 (Mention) 数量的分桶值
NUM_AGR	如果 π 和 α 在单复数形式上符合则为 T, 否则为 F
	如果 π 或者 α 的单数复数形式不能确定则为 UNK
GEN_AGR	如果 π 和 α 性别一致则为 T, 否则为 F
	如果 π 或者 α 的性别不能确定则为 UNK

意表述 π , 其正样本选取和 π 存在指代关系的距离最近的一个先行词。其负样本选取策略是：在选定正样本后，以正样本为中心，选取窗口大小为 4 个句子内的所有和 π 不具有指代关系的表述作为负样本，其中，4 个句子包括 π 所在的句子、 π 所在句的前一个句子、 π 所在句的后两个句子。在训练过程中，模型需要最大化正样本的条件概率 $P_r(\alpha_i | \pi)$ ，而负样本则作为分子中的项被计算在损失函数中。

在测试时，考虑到大部分指代为局部指代，并为了节约测试时间，RK 算法只选取指代词 π 所在的句子及所在句之前的 3 个句子内的表述作为候选。模型基于所有候选词，计算每个词被选为和 π 构成指代关系的先行词的概率并选取概率最高的作为输出结果。

2. 基于循环神经网络的端到端的表述排序指代消解

Lee 等人提出了端到端的指代消解模型 E2E-COREF^[30]，同样基于表述排序的方法实现。E2E-COREF 在训练时同时学习判断每个片段 (span) 是否为实体表述、及优化对实体表述的指代聚类。

具体来说，对于每个片段 i ，模型的目标是在所有候选先行词中选出一个其指代的先行词 y_i 。其中，先行词的候选集合为 $\mathcal{Y}(i) = \{\epsilon, 1, \dots, i - 1\}$ ，包括一个虚先行词 ϵ 及所有在 i 之前的片段

(需要注意的是，这里的片段可能不是实体表述)。当模型选择虚先行词 ϵ 作为输出时，可能对应两种情况：(1) 该片段 i 不是实体表述；(2) 该片段 i 是实体表述，但不指代在其之前的任一个片段（例如，可能是该实体在文中的第一次提及）。由此，根据对每个片段得到的先行词预测，可以构建出整个文本中的指代集合。

与 RK 算法相似，E2E-COREF 对每个候选先行词，计算其和片段 i 为指代关系的条件概率：

$$P(y_i) = \frac{\exp(s(i, y_i))}{\sum_{y' \in \mathcal{Y}(i)} \exp(s(i, y'))} \quad (5.27)$$

其中 $s(i, j)$ 是表示片段 i 和片段 j 之间存在指代关系的分数，这一分数与三个因素相关：(1) $s_m(i)$ ：片段 i 是否为实体表述；(2) $s_m(j)$ ：片段 j 是否为实体表述；(3) $s_a(i, j)$ ：片段 j 是否为 i 的先行词：

$$s(i, j) = \begin{cases} 0 & j = \epsilon \\ s_m(i) + s_m(j) + s_a(i, j) & j \neq \epsilon \end{cases} \quad (5.28)$$

通过将虚先行词的分数设为 0，当模型预测任意非虚先行词的分数为正时，则可以选出分数最高的先行词预测；当模型预测所有非虚先行词的分数都为负时，则输出虚先行词。接下来我们将介绍模型如何计算以上三个分数。

针对片段表示编码表示，E2E-COREF 首先基于双向 LSTM 网络和注意力机制编码片段表示，其网络结构如图5.16所示。对于每个输入句子，其输入表示 $\mathbf{x}_1, \dots, \mathbf{x}_T$ 由预训练词向量及对字符的 1 维卷积组成。模型首先基于双向 LSTM 网络得到每个词的上下文表示 $\mathbf{x}_t^* = [\mathbf{h}_{t,1}, \mathbf{h}_{t,-1}]$ ，其中 \mathbf{x}_t^* 是 LSTM 的前向输出表示 $\mathbf{h}_{t,1}$ 和反向输出表示 $\mathbf{h}_{t,-1}$ 的拼接（关于 LSTM 网络的详细介绍可见。。）。基于词表示 \mathbf{x}_t^* ，E2E-COREF 通过注意力机制判断片段中每个词的重要性，找出可能的关键词并给予更高权重，适用加权计算得到关键片段表示 $\hat{\mathbf{x}}_i$ ：

$$\begin{aligned} \alpha_t &= \mathbf{w}_\alpha \cdot \text{FFNN}_\alpha(\mathbf{x}_t^*) \\ a_{i,t} &= \frac{\exp(\alpha_t)}{\sum_{k=\text{START}(i)}^{\text{END}(i)} \exp(\alpha_k)} \\ \hat{\mathbf{x}}_i &= \sum_{t=\text{START}(i)}^{\text{END}(i)} a_{i,t} \cdot \mathbf{x}_t \end{aligned} \quad (5.29)$$

其中，FFNN 表示前馈神经网络， \cdot 表示点积运算。

对于每个片段 i ，其片段表示由其首尾词表示、关键片段表示及一个表示片段长度的特征向量 $\phi(i)$ 的拼接构成：

$$\mathbf{g}_i = [\mathbf{x}_{\text{START}(i)}^*, \mathbf{x}_{\text{END}(i)}^*, \hat{\mathbf{x}}_i, \phi(i)] \quad (5.30)$$

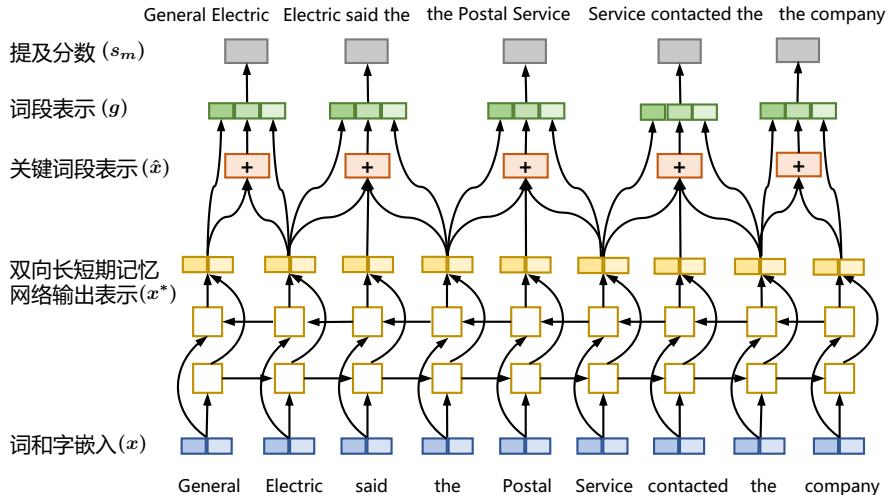


图 5.16 E2E-COREF 基于双向 LSTM 的片段表示编码

针对分数计算，E2E-COREF 基于上述得到的片段表示，使用前馈神经网络计算 5.28 中的各项分数，其网络结构如图 5.17 所示。分数具体计算公式如下：

$$\begin{aligned} s_m(i) &= \mathbf{w}_m \cdot \text{FFNN}_m(\mathbf{g}_i) \\ s_a(i, j) &= \mathbf{w}_a \cdot \text{FFNN}_a([\mathbf{g}_i, \mathbf{g}_j, \mathbf{g}_i \circ \mathbf{g}_j, \phi(i, j)]) \end{aligned} \quad (5.31)$$

其中 \cdot 表示点积运算， \circ 表示逐项乘积运算， $[.]$ 表示向量拼接。可以看到，片段的实体表述分数 $s_m(i)$ 与片段表示相关，而片段对的指代关系预测分数 $s_a(i, j)$ 和两个片段表示、两个片段表示的乘积、及一个与话语主体、文本类别（从元数据中获得）和片段对距离相关的特征表示 $\phi(i, j)$ 相关。

在训练阶段，对于每个片段 i ，E2E-COREF 希望最大化与其具有正确指代关系的所有片段的条件概率 $P(y_i)$ 。假设 $\text{GOLD}(i)$ 为与片段 i 的具有正确指代关系的片段集合， $\mathcal{Y}(i)$ 为我们前面定义的先行词候选集合，则基于交叉熵损失，模型希望最小化的损失可以表示为：

$$-\sum_{i=2}^N \log \sum_{\hat{y} \in \mathcal{Y}(i) \cap \text{GOLD}(i)} P(\hat{y}) \quad (5.32)$$

需要注意的是，如果片段 i 不和任何片段构成指代关系，则其正确指代片段集合 $\text{GOLD}(i) = \{\epsilon\}$ 。

在测试时，考虑到对所有片段组合计算的效率问题，E2E-COREF 制定了以下策略：(1) 只考虑词数小于等于 L 的片段；(2) 在计算得到每个片段的实体表述分数 $s_m(i)$ 后，只保留分数最高的 γT 个片段进行后续计算；(3) 对于每个保留的片段，只考虑其最近的 K 个先行词候选进行片段对的分数计算。基于上述策略和公式 5.27，E2E-COREF 能够解码得到最可能的指代关系分布。

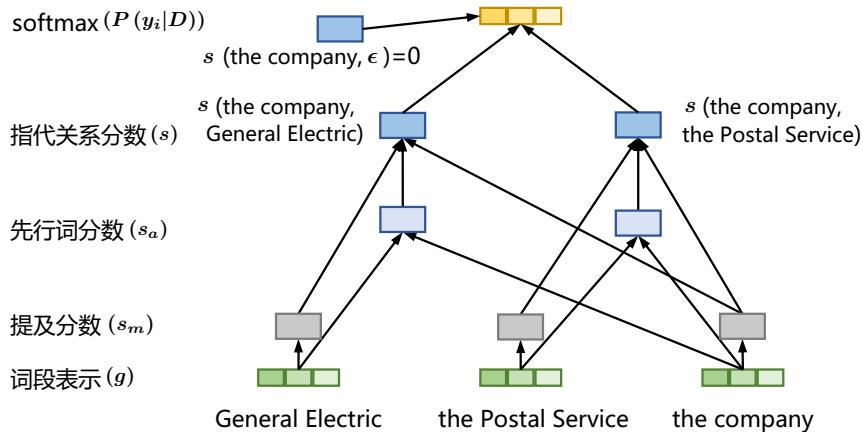


图 5.17 E2E-COREF 的分数计算

5.4.3 基于实体的指代消解

基于表述对和基于表述排序的两种指代消解算法旨在将一个表述与其所指代的一个表述相对应，通常只关注局部的指代信息。基于实体的指代消解则认为将单个表述归类至其指代的实体（通常对应一个表述的等价类）能利用实体级别的全局信息，因此能更好地实现指代消解任务。

基于实体的指代消解和基于表述的指代消解算法相似，区别在于基于表述的方法将当前表述分配到一个先行的表述，而基于实体的方法将当前表述分配到先行的实体（表述等价类）上。基于实体的指代消解同样可以分为基于实体-表述的方法和基于实体排序的方法。在本节中，我们将介绍基于特征工程的实体-表述指代消解和基于循环神经网络的实体排序指代消解。

1. 基于特征工程和 SVM 分类器的实体-表述指代消解

Rahman 等人提出了基于实体-表述的指代消解算法 Entity-mention-coref^[31]。基于实体-表述的指代消解算法和 5.4.1 节中所介绍的基于表述对的指代消解算法相似，都是通过训练二分类器对每个表述进行分类。不同之处在于，对于任意表述 m_i ，基于表述对的指代消解算法仅关注其是否和某一候选先行词 m_j 为指代关系，并训练一个二分类器计算 m_i 和 m_j 为指代关系的分数 $s(m_j, m_i)$ 。而基于实体-表述的指代消解算法则关注表述 m_i 是否属于某个表述集群 c_j ，该表述集群由指代同一实体的所有先行词构成。同样地，Entity-mention-coref 通过训练一个 SVM 二分类器计算 m_i 属于表述集群 c_j 的分数 $s(c_j, m_i)$ ，其中 $c_j \in \mathcal{C}(i)$ ， $\mathcal{C}(i)$ 代表在 m_i 之前的所有实体（表述集群）的集合。

为了更好地表示输入数据，Entity-mention-coref 中所构建的特征包含两类：(1) 描述待分类表述 m_i 的表述级别特征，和 5.4.1 节基于表述对的指代消解方法所构建的表述级别特征相似，包括其词性特征等；(2) 实体级别的特征，用于描述待分类表述 m_i 和候选实体 c_j 的关系，包括实体

级别的性别、数字、语意相符程度；实体级别的距离特征；实体级别的字符串关系特征等。部分构建的特征如表5.2所示。

表 5.2 Entity-mention-coref 使用的部分特征

描述待分类表述 m_i 的表述级别特征	
NUMBER_2	SINGULAR 或 PLURAL, 根据词典确定
GENDER_2	MALE, FEMALE, NEUTER, or UNKNOWN, 根据常见人名列表确认
PRONOUN_2	如果 m_k 是代词则为 Y, 否则为 N
待分类表述 m_i 和其候选先行词 m_k 的关系特征	
HEAD_MATCH	如果表述具有相同的中心名词则为 C, 否则为 I
STR_MATCH	如果表述具有相同的字符串则为 C, 否则为 I
SUBSTR_MATCH	如果一个表述是另一个的子字符串则为 C, 否则为 I
PRO_STR_MATCH	如果两个表示都是代词并且相同则为 C, 否则为 I
描述待分类表述 m_i 和其候选先行词 m_k 的关系的额外特征	
NUMBER'	m_j 和 m_k 的 NUMBER_2 特征合并
GENDER'	m_j 和 m_k 的 GENDER_2 特征合并
PRONOUN'	m_j 和 m_k 的 PRONOUN_2 特征合并

在训练时，对于每个表述 m_i ，Entity-mention-coref 使用 m_i 之前的实体分别组成正、负训练样本，其中，正样本由 m_i 和其所属的实体 c_j 组成；负样本由 m_j 到其所属实体距离 m_i 最近的先行词 m_j （也即实体 c_j 的最后一个表述）之间的表述和 m_i 组成。

例如：[Barack Obama]¹ nominated [Hillary Rodham Clinton]² as [[his]³ secretary of state]³ on [Monday]⁴. [He]¹...

其中，每个表述的下标代表其出现的次序，上标代表其所属的实体。当对表述“He”进行分类时，将产生三条训练样本： $I(\{Monday\}, He)$ ， $I(\{secretaryofstate\}, He)$ 和 $I(\{BarackObama, his\}, He)$ 。其中，前两条样本为负样本，最后一条为正样本。

在测试时，同样考虑 m_i 之前出现的所有实体，并基于??中描述的最近原则，将 m_i 分配至与其距离最近的被分类器判别为存在指代关系的实体。相反，如果 m_i 和之前的所有实体都不存在指代关系，则其将被认为是一个新的实体。

2. 基于循环神经网络的实体排序指代消解

Wiseman 等人提出了基于循环神经网络的实体排序指代消解模型 Global-rank^[32]。基于实体排序的指代消解方法和本章第5.4.2节中所介绍的基于表述排序的方法相似。具体来说，对于当前待分类表述 x_n ，模型通过对其之前出现的所有实体（表述集群） $\{X^{(m)}\}_{m=1}^M$ 计算分数并排序，选出 x_n 所属的集群。

由于 Global-rank 是基于表述集群的算法，首先定义集群的相关符号： $X^{(m)}$ 为第 m 个表述集群， $X_j^{(m)}$ 为第 m 个集群中的第 j 个表述，其中表述的排序由其在文档中出现的顺序确定。由于一个有效的文档表述聚类会将每个表述都分入一个确定的集群，定义一组表述-集群映射 $\mathbf{z} \in \{1, \dots, M\}^N$ ，当 x_n 属于第 m 个集群时 $z_n = m$ 。

在本章第5.4.2节中，我们介绍了基于表述排序的指代消解，其基本思路是对每个 x_n 和候选先行词 $y \in \mathcal{Y}(x_n)$ 计算其为指代关系的分数 $f(x_n, y)$ ，其中 $\mathcal{Y}(x_n) = \{1, \dots, n-1, \epsilon\}$ 为 x_n 的候选先行词集合， ϵ 为虚先行词，表示当前表述为该实体在文中的第一次提及。这样可以使得基于表述排序的算法能够高效计算和解码。然而，这种方式使用单个指代代表整个集群，只能计算局部的表述分数，无法利用整个集群的全局信息。因此，Global-rank 在局部表述分数 $f(x_n, y)$ 的基础上，增添了一项全局实体-表述分数 $g(x_n, y, z_{1:n-1})$ 的计算，其中 $z_{1:n-1}$ 代表当前表述 x_n 之前所有表述所属的集群的映射。具体实现时，为了保留基于表述排序的方法的优势，并简化测试过程，Global-rank 在解码时同时计算局部和全局的分数，并寻找使该分数最大的集群分配方式：

$$\arg \max_{y_1, \dots, y_N} \sum_{n=1}^N f(x_n, y_n) + g(x_n, y_n, z_{1:n-1}) \quad (5.33)$$

接下来我们将分别介绍局部表述分数 $f(x_n, y)$ 和全局实体-表述分数 $g(x_n, y, z_{1:n-1})$ 的计算。

局部表述分数计算和第5.4.2节中介绍的表述排序分数计算相似，Global-rank 首先基于文献[33]中定义的特征映射 $\phi_a(x_n) : \mathcal{X} \rightarrow \{0, 1\}^F$ 和 $\phi_p(x_n, y) : (\mathcal{X}, \mathcal{X}) \rightarrow \{0, 1\}^F$ 将表述 x_n 和表述对 (x_n, y) 分别表示成表述和表述对级别的特征向量，接着使用非线性特征映射 \mathbf{h}_a 和 \mathbf{h}_p 将该特征向量映射到连续的特征空间：

$$\begin{aligned} \mathbf{h}_a(x_n) &= \tanh(\mathbf{W}_a \phi_a(x_n) + \mathbf{b}_a) \\ \mathbf{h}_p(x_n, y) &= \tanh(\mathbf{W}_p \phi_p(x_n, y) + \mathbf{b}_p) \end{aligned} \quad (5.34)$$

则局部表述分数计算定义如下：

$$f(x_n, y) = \begin{cases} \mathbf{u}^\top [\mathbf{h}_a(x_n), \mathbf{h}_p(x_n, y)] + u_0 & \text{if } y \neq \epsilon \\ \mathbf{v}^\top \mathbf{h}_a(x_n) + v_0 & \text{if } y = \epsilon \end{cases} \quad (5.35)$$

在计算全局的实体-表述分数之前，首先需要对实体集群级别的表示进行计算。Global-rank 使用 RNN 对每个集群所包含的表述进行依次编码，从而计算该实体集群的整体表示。具体来说，首先将表述表示成和公式5.34相似的形式：

$$\mathbf{h}_c(x_n) = \tanh(\mathbf{W}_c \phi_a(x_n) + \mathbf{b}_c) \quad (5.36)$$

其中 \mathbf{W}_c 和 \mathbf{b}_c 为全局分数计算所对应的表示参数。

接着，对于集群 m 中的第 j 个表述，基于 RNN 的表示计算可以表示成：

$$\mathbf{h}_j^{(m)} \leftarrow \text{RNN}(\mathbf{h}_c(X_j^{(m)}), \mathbf{h}_{j-1}^{(m)}; \boldsymbol{\theta}) \quad (5.37)$$

在对全局实体-表述分数进行计算时，使用 $\mathbf{h}_{<n}^{(z_y)}$ 表示集群 z_y 在对 x_n 之前所有的表述依次编码后得到的集群表示，并将全局实体-表述分数定义成和局部表述分数计算相似的形式：

$$g(x_n, y, \mathbf{z}_{1:n-1}) = \begin{cases} \mathbf{h}_c(x_n)^\top \mathbf{h}_{<n}^{(z_y)} & \text{if } y \neq \epsilon \\ \text{NA}(x_n) & \text{if } y = \epsilon \end{cases} \quad (5.38)$$

其中，NA 对该表述为首次出现的情况进行了分别计算：

$$\text{NA}(x_n) = \mathbf{q}^\top \tanh(\mathbf{W}_s[\phi_a(x_n), \sum_{m=1}^M \mathbf{h}_{<n}^{(m)}] + \mathbf{b}_s) \quad (5.39)$$

在训练时，每个表述所属的实体由训练集的标注所给定，但每个表述所指代的先行词可以有多个。因此，Global-rank 对每个表述，选取使其分数最高的先行词作为隐式目标先行词，并定义了如下基于间隔（margin）的目标函数：

$$\sum_{n=1}^N \max_{\hat{y} \in \mathcal{Y}(x_n)} \Delta(x_n, \hat{y})(1 + f(x_n, \hat{y}) + g(x_n, \hat{y}, \mathbf{z}^{(o)}) - f(x_n, y_n^l), g(x_n, y_n^l, \mathbf{z}^{(o)})) \quad (5.40)$$

其中，当 x_n 为前指项时，隐式目标先行词定义为使局部和全局分数之和最高的先行词，否则为 ϵ ：

$$y_n^l = \arg \max_{y \in \mathcal{Y}(x_n): z_y^{(o)} = z_n^{(o)}} f(x_n, y) + g(x_n, y, \mathbf{z}^{(o)}) \quad (5.41)$$

$\Delta(x_n, \hat{y})$ 表示针对文献 [34] 定义的不同错误类型“假链接(False link)”、“假新实体(False new)”、“错误链接(Wrong link)”定义不同的损失权重 $(\alpha_1, \alpha_2, \alpha_3)$ ，其中，“假链接”表示预测当前表述为前指，实际为新实体的错误；“假新实体”表示预测表述为新实体，实际为前指；“错误链接”为预测了错误的指代先行词。

Global-rank 的解码算法如算法 5.2 所示。对于每个待分类表述 x_n ，首先计算得到使得局部和全局分数之和最高的先行词 y^* ，并得到 y^* 所属的集群 m 。当 $y^* = \epsilon$ 时，表示当前表述不属于任何集群，则建立一个新的集群。接着，更新集群 m ，集群映射及集群 m 的表示，以用于下次解码计算。对所有表述 x_n 遍历分类后，返还所有集群。

代码 5.2: Global-rank 解码算法

输入: 待解码文本序列 (x_1, \dots, x_N)
 输出: 实体集群 $X^{(1)}, \dots, X^{(M)}$

```

// 初始化
foreach  $X^{(i)}$  do
|  $X^{(i)} \leftarrow []$ ;           // 初始化每个集群  $X^{(i)}$  为空列表;
end
foreach  $h_0^{(i)}$  do
|  $h_0^{(i)} \leftarrow \mathbf{0} \in \mathbb{R}^D$ ;    // 初始化每个集群的 RNN 初始输入  $h_0^{(0)}$  为 0 向量;
end
 $z \leftarrow \mathbf{0}$ ;                  // 初始化集群映射向量;
 $M \leftarrow 0$ ;                    // 初始化集群计数;
// 解码过程
for  $n = 2$  to  $N$  do
|  $y^* \leftarrow \arg \max_{y \in \mathcal{Y}(x_n)} f(x_n, y) + g(x_n, y, z_{1:n-1})$ ; // 计算使局部和全局分数之和最高的先行
  词;
|  $m \leftarrow z_{y^*}$ ;                // 得到  $y^*$  所属的集群;
| if  $y^* = \epsilon$  then
| | // 建立一个新的集群
| |  $M \leftarrow M + 1$ ;
| |  $m \leftarrow M$ ;
| end
|  $X^{(m)} \leftarrow X^{(m)} + [x_n]$ ;
|  $z_n \leftarrow m$ ;
|  $h^{(m)} \leftarrow \text{RNN}(h_c(x_n), h^{(m)})$ ;          // 更新集群  $m$  的表示;
end
return  $X^{(1)}, \dots, X^{(M)}$ 

```

5.5 延伸阅读

5.6 习题

参考文献

- [1] De Beaugrande R A, Dressler W U. Introduction to text linguistics: volume 1[M]. longman London, 1981.
- [2] Halliday M A K, Hasan R. Cohesion in english[M]. Routledge, 2014.
- [3] 苗兴伟, 张蕾. 汉语语篇分析[M]. 外语教学与研究出版社, 2021.
- [4] Van Dijk T A. News analysis[J]. Case Studies of International and National News in the Press. New Jersey: Lawrence, 1988.
- [5] Mann W C, Thompson S A. Rhetorical structure theory: A theory of text organization[M]. University of Southern California, Information Sciences Institute Los Angeles, 1987.
- [6] Hoey M. Textual interaction: An introduction to written discourse analysis[M]. Psychology Press, 2001.
- [7] Soricut R, Marcu D. Sentence level discourse parsing using syntactic and lexical information[C/OL]// Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics. 2003: 228-235. <https://aclanthology.org/N03-1030>.
- [8] Magerman D M. Statistical decision-tree models for parsing[J]. arXiv preprint cmp-lg/9504030, 1995.
- [9] Carlson L, Marcu D, Okurovsky M E. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory[C/OL]//Proceedings of the Second SIGdial Workshop on Discourse and Dialogue. 2001. <https://aclanthology.org/W01-1605>.
- [10] Wang Y, Li S, Yang J. Toward fast and accurate neural discourse segmentation[C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 962-967. <https://aclanthology.org/D18-1116>. DOI: 10.18653/v1/D18-1116.

- [11] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[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: 2227-2237. <https://aclanthology.org/N18-1202>. DOI: 10.18653/v1/N18-1202.
- [12] Prasad R, Dinesh N, Lee A, et al. The Penn Discourse TreeBank 2.0.[C/OL]//Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08). Marrakech, Morocco: European Language Resources Association (ELRA), 2008. http://www.lrec-conf.org/proceedings/lrec2008/pdf/754_paper.pdf.
- [13] Hernault H, Prendinger H, du Verle D A, et al. Hilda: A discourse parser using support vector machine classification[J]. Dialogue & Discourse, 2010, 1(3):1-33.
- [14] Feng V W, Hirst G. Text-level discourse parsing with rich linguistic features[C/OL]//Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Jeju Island, Korea: Association for Computational Linguistics, 2012: 60-68. <https://aclanthology.org/P12-1007>.
- [15] Li J, Li R, Hovy E. Recursive deep models for discourse parsing[C/OL]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014: 2061-2069. <https://aclanthology.org/D14-1220>. DOI: 10.3115/v1/D14-1220.
- [16] Webber B. D-ltag: extending lexicalized tag to discourse[J]. Cognitive Science, 2004, 28(5):751-779.
- [17] Pitler E, Nenkova A. Using syntax to disambiguate explicit discourse connectives in text[C/OL]// Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. Suntec, Singapore: Association for Computational Linguistics, 2009: 13-16. <https://aclanthology.org/P09-2004>.
- [18] Marcinkiewicz M A. Building a large annotated corpus of english: The penn treebank[J]. Using Large Corpora, 1994, 273.
- [19] Pitler E, Raghupathy M, Mehta H, et al. Easily identifiable discourse relations[C/OL]//Coling 2008: Companion volume: Posters. Manchester, UK: Coling 2008 Organizing Committee, 2008: 87-90. <https://aclanthology.org/C08-2022>.
- [20] Rutherford A, Demberg V, Xue N. A systematic study of neural discourse models for implicit discourse relation[C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. 2017: 281-291.

- [21] Zhang B, Su J, Xiong D, et al. Shallow convolutional neural network for implicit discourse relation recognition[C/OL]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015: 2230-2235. <https://aclanthology.org/D15-1266>. DOI: 10.18653/v1/D15-1266.
- [22] Ji Y, Haffari G, Eisenstein J. A latent variable recurrent neural network for discourse-driven language models[C/OL]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, California: Association for Computational Linguistics, 2016: 332-342. <https://aclanthology.org/N16-1037>. DOI: 10.18653/v1/N16-1037.
- [23] Shi W, Demberg V. Next sentence prediction helps implicit discourse relation classification within and across domains[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: 5790-5796. <https://aclanthology.org/D19-1586>. DOI: 10.18653/v1/D19-1586.
- [24] Ji Y, Cohn T, Kong L, et al. Document context language models[J]. arXiv preprint arXiv:1511.03962, 2015.
- [25] Bengtson E, Roth D. Understanding the value of features for coreference resolution[C/OL]// Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. Honolulu, Hawaii: Association for Computational Linguistics, 2008: 294-303. <https://aclanthology.org/D08-1031>.
- [26] Soon W M, Ng H T, Lim D C Y. A machine learning approach to coreference resolution of noun phrases[J/OL]. Computational Linguistics, 2001, 27(4):521-544. <https://aclanthology.org/J01-4004>. DOI: 10.1162/089120101753342653.
- [27] Ng V, Cardie C. Improving machine learning approaches to coreference resolution[C/OL]// Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, 2002: 104-111. <https://aclanthology.org/P02-1014>. DOI: 10.3115/1073083.1073102.
- [28] Clark K, Manning C D. Improving coreference resolution by learning entity-level distributed representations[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: 643-653. <https://aclanthology.org/P16-1061>. DOI: 10.18653/v1/P16-1061.

- [29] Denis P, Baldridge J. A ranking approach to pronoun resolution.[C]//IJCAI: volume 158821593. 2007.
- [30] Lee K, He L, Lewis M, et al. End-to-end neural coreference resolution[C/OL]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, 2017: 188-197. <https://aclanthology.org/D17-1018>. DOI: 10.18653/v1/D17-1018.
- [31] Rahman A, Ng V. Supervised models for coreference resolution[C/OL]//Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. Singapore: Association for Computational Linguistics, 2009: 968-977. <https://aclanthology.org/D09-1101>.
- [32] Wiseman S, Rush A M, Shieber S M. Learning global features for coreference resolution[C/OL]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, California: Association for Computational Linguistics, 2016: 994-1004. <https://aclanthology.org/N16-1114>. DOI: 10.18653/v1/N16-1114.
- [33] Wiseman S, Rush A M, Shieber S, et al. Learning anaphoricity and antecedent ranking features for coreference resolution[C/OL]//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). Beijing, China: Association for Computational Linguistics, 2015: 1416-1426. <https://aclanthology.org/P15-1137>. DOI: 10.3115/v1/P15-1137.
- [34] Durrett G, Klein D. Easy victories and uphill battles in coreference resolution[C/OL]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics, 2013: 1971-1982. <https://aclanthology.org/D13-1203>.

索引

- Anaphor, 23
- anaphora, 23
- Antecedent, 23
- coreference, 23
- Coreference Resolution, 23
- Discourse, 1
- Discourse Segmentation, 9
- Explicit Discourse Relation, 17
- Implicit Discourse Relation, 18
- Superstructure, 6
- Textual Pattern, 8
- 下指照应, 3
- 修辞结构理论, 7
- 先行词, 23
- 共指, 23
- 内指照应, 3
- 回指, 23
- 回指照应, 3
- 外指照应, 3
- 指代消解, 23
- 搭配关系, 2
- 显式篇章关系, 17
- 替代, 3
- 照应, 2
- 照应词, 23
- 省略, 3
- 篇章, 1
- 篇章超级结构, 6
- 衔接, 2