17. 信息提取

I am the very model of a modern Major-General,
I've information vegetable, animal, and mineral,
I know the kings of England, and I quote the fights historical
From Marathon to Waterloo, in order categorical...

Gilbert and Sullivan, *Pirates of Penzance*

假设您是一家跟踪航空公司股票的投资公司的分析师。您的任务是确定航空公司宣布的票价上涨与次日股票走势之间的关系(如果有的话)。股票价格的历史数据很容易获得，但航空公司的公告呢?您至少需要知道航空公司的名称、建议涨价的性质、公告的日期，以及其他航空公司可能的回应。幸运的是，这些都可以在像这样的新闻文章中找到:

Citing high fuel prices, United Airlines said Friday it has increased fares by $6 per round trip on flights to some cities also served by lowercost carriers. American Airlines, a unit of AMR Corp., immediately matched the move, spokesman Tim Wagner said. United, a unit of UAL Corp., said the increase took effect Thursday and applies to most routes where it competes against discount carriers, such as Chicago to Dallas and Denver to San Francisco.

本章介绍了从文本中提取有限种类语义内容的技术。这个**信息提取(information extraction，IE)**将嵌入文本中的非结构化信息转化为结构化数据，例如填充关系数据库以实现进一步处理。

我们从**关系提取(relation extraction)**的任务开始：在文本实体之间查找和分类语义关系。这些通常是二元关系，例如子女关系，就业关系，部分整体关系和地理空间关系。关系提取与填充关系数据库有着紧密的联系。确实，**知识图谱(knowledge graphs)**，结构化关系知识的数据集，是搜索引擎向用户提供信息的常见方式。

接下来，我们将讨论三个与事件相关的任务。**事件提取(event extraction)**是寻找这些实体参与的事件，例如，在我们的示例文本中，United 和 American 的票价增加以及 aid 和 cite 的报告事件。需要使用**事件共指(event coreference)**(第 22 章)来确定文本中提到的哪个事件引用的是同一个事件;在我们正在运行的示例中，increase 的两个实例和短语 the move 都指向同一个事件。

为了弄清文本中的事件何时(when)发生，我们提取**时间表达式(temporal expressions)**，例如一周中的几天(Friday 和 Thursday)，**相对表达式(relative expressions)**，例如 two days from now 或者 next year，以及时间诸如 3:30 PM，这些表达式必须标准化为特定的日历日期或一天中的时间，以便及时定位事件。在我们的示例任务中，这将使我们能够将 Friday 与 United's 宣布时间连接起来，将 Thursday 与前一天的加价连接起来，并产生一个时间表，在该时间表中，United's 宣布会跟随加价，而 American's 宣布会跟随这两个事件。

最后，许多文本描述了反复出现的陈规定型事件或情况。**模板填充(template filling)**的任务是在文档中查找此类情况并填写模板槽。这些槽填充符可以包含直接从文本中提取的文本段，也可以包括通过额外处理从文本元素中推断出的诸如时间、数量或本体(ontology)实体之类的概念。

我们的航空公司文字是这种陈规定型情况的一个例子，因为航空公司经常提高票价，然后等待竞争对手是否跟进。在这种情况下，我们可以将 United 确定为最初提高票价的领头航空公司，将票价定为$6，将 Thursday 为加价日期，将 American 定为紧随其后的航空公司，从而产生如下所示的填充模板。

```
FARE-RAISE ATTEMPT: [ LEAD AIRLINE:    UNITED AIRLINES
                      AMOUNT:          $6
                      EFFECTIVE DATE:  2006-10-26
                      FOLLOWER:        AMERICAN AIRLINES ]
```

17.1. 关系提取

让我们假设我们已经在示例文本中检测到了命名实体(可能使用了第 8 章的技术)，并想要辨别存在于检测到的实体之间的关系:

> Citing high fuel prices, [ORG **United Airlines**] said [TIME **Friday**] it has increased fares by [MONEY **$6**] per round trip on flflights to some cities also served by lower-cost carriers. [ORG **American Airlines**], a unit of [ORG **AMR Corp.**], immediately matched the move, spokesman [PER **Tim Wagner**] said. [ORG **United**], a unit of [ORG **UAL Corp.**], said the increase took effect [TIME **Thursday**] and applies to most routes where it competes against discount carriers, such as [LOC **Chicago**] to [LOC **Dallas**] and [LOC **Denver**] to [LOC **San Francisco**].

例如，文本告诉我们，Tim Wagner 是 American Airlines 的发言人，United 是 UAL Corp.的子公司，而 American 是 AMR 的子公司。这些二元关系是更一般的关系的实例，例如在新闻风格的文本中相当常见的部分(part-of)或雇佣(employs)关系。图 17.1 列出了 ACE 关系抽取评估中使用的 17 种关系，图 17.2 给出了一些样本关系。我们还可以提取更多特定于领域的关系，比如航线的概念。例如，从本文中我们可以得出结论，United 有飞往 Chicago、Dallas、Denver、和 San Francisco 航线。

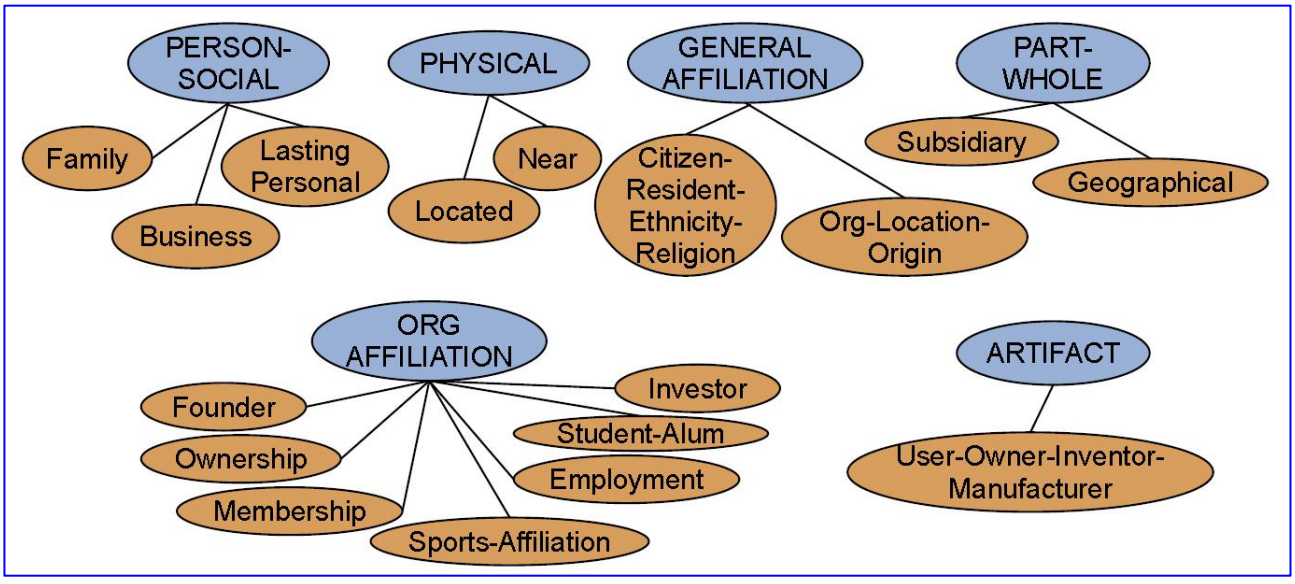

图 17-1：ACE 关系提取任务中使用的 17 个关系

Relations	Types	Examples
Physical-Located	PER-GPE	He was in Tennessee
Part-Whole-Subsidiary	ORG-ORG	XYZ, the parent company of ABC
Person-Social-Family	PER-PER	Yoko's husband John
Org-AFF-Founder	PER-ORG	Steve Jobs, co-founder of Apple...

图 17-2：带有示例的语义关系及其所涉及的命名实体类型

这些关系与我们在第 15 章中介绍的用于奠定逻辑形式含义基础的模型理论概念非常吻合。也就是说，一个关系由领域元素上的一组有序元组组成。在大多数标准的信息提取应用程序中，领域元素对应于文本中出现的命名实体，对应于共指消解产生的底层实体，或者对应于从领域本体中选择的实体。图 17.3 显示了一组实体和关系的基于模型的视图，这些实体和关系可以从我们正在运行的示例中提取出来。

请注意这个模型理论视图是如何包含 NER 任务的;命名实体识别对应于一类一元(unary)关系的识别。

也为许多其他领域定义了关系集。例如 UMLS，来自美国国家医学图书馆的统一医学语言系统，其网络定义了 134 个广泛的主题类别、实体类型和实体之间的 54 种关系，例如:

Domain	$\mathcal{D}$ = {a,b,c,d,e,f,g,h,i }
United, UAL, American Airlines, AMR	a,b,c,d
Tim Wagner	e
Chicago, Dallas, Denver, and San Francisco	f,g,h,i
Classes	
United, UAL, American, and AMR are organizations	Org = {a,b,c,d}
Tim Wagner is a person	Pers= {e}
Chicago, Dallas, Denver, and San Francisco are places	Loc = {f,g,h,i}
Relations	
United is a unit of UAL American is a unit of AMR	PartOf = {<a,b> , <c,d>}
Tim Wagner works for American Airlines	OrgAff = {<c,e>}
United serves Chicago, Dallas, Denver, and San Francisco	Serves= {<a,f >, <a,g>, <a,h>, <a,i>}

图 17-3：示例文本中基于模型的关系和实体视图

Entity	Relation	Entity
Injury	disrupts	Physiological Function
Bodily Location	location-of	Biologic Function
Anatomical Structure	part-of	Organism
Pharmacologic Substance	causes	Pathological Function
Pharmacologic Substance	treats	Pathologic Function

给出这样一个医学的句子:

(17.1) Doppler echocardiography can be used to diagnose left anterior descending artery stenosis in patients with type 2 diabetes

因此，我们可以提取 UMLS 关系:

Echocardiography, Doppler Diagnoses *Acquired stenosis*

Wikipedia 还提供了大量的关系，这些关系是从与某些 Wikipedia 文章相关的**信息框(infoboxes)**，结构化表格中得出的。例如，斯坦福大学的 Wikipedia 信息框包含结构化事实，例如 state="California"或 President="Marc Tessier-Lavigne"。这些事实可以转化为诸如 president-of 或者 located-in 之类的关系。或在称为**资源描述框架(Resource Description Framework，RDF)**的元语言(metalanguage)中建立关系。RDF **三元组(triple)**是实体关系实体的元组，称为**主-谓-宾(subject-predicate-object)**表达式。这是一个示例 RDF 三元组：

subject	predicate	object
Golden Gate Park	location	San Francisco

例如，众包的(crowdsourced) DBpedia (Bizer 等人，2009)是从 Wikipedia 派生出来的本体，包含超过 20 亿个 RDF 三元组。另一个来自维基百科信息箱的数据集 Freebase (Bollacker 等人，2008)，现在是 Wikidata Freebase 的一部分(Vrandecic 和 Krotzsch, 2014)，具有人们和他们的国籍、地点以及他们所包含的其他地点之间的关系。

WordNet 或其他本体提供有用的本体关系，表达词或概念之间的层次关系。例如，WordNet 在类之间有 is-a 或**上位词(hypernym)**关系：

Giraffe is-a ruminant is-a ungulate is-a mammal is-a vertebrate ...

(长颈鹿— 反刍动物 — 有蹄动物 — 哺乳动物 — 脊椎动物…)

WordNet 在个体和类别之间也有实例(Instance-of)关系，例如，San Francisco 与 city 之间的实例关系。提取这些关系是扩展或构建本体的重要步骤。

最后，有大量的数据集，其中包含利用它们的关系来做手工标记的句子，用于训练和测试关系提取器。TACRED 数据集(Zhang 等人，2017)包含关于特定人或组织的关系三元组的 106,264 个例子，以从年度 TAC 知识库人口(TAC KBP)挑战中抽取的新闻和 web 文本的句子标注。TACRED 包含 41 种关系类型(如

per:出生城市，org:子公司，org:成员，per:配偶)，加上一个无关系(no relation)标记;示例如图 17.4 所示(Zhang 等人，2017)。大约 80%的例子被标注为没有关系;有足够的负面数据对于训练有监督的分类器是很重要的。

Example	Entity Types & Label
Carey will succeed Cathleen P. Black, who held the position for 15 years and will take on a new role as chairwoman of Hearst Magazines, the company said.	PERSON/TITLE Relation: per:title
Irene Morgan Kirkaldy, who was born and reared in Baltimore, lived on Long Island and ran a child-care center in Queens with her second husband, Stanley Kirkaldy.	PERSON/CITY Relation: per:city of birth
Baldwin declined further comment, and said JetBlue chief executive Dave Barger was unavailable.	Types: PERSON/TITLE Relation: no relation

图 17-4：来自 TACRED 数据集的例句和标签

SemEval 2010 任务 8 也生成了一个标准数据集，用于检测名词性词(nominals)之间的关系(Hendrickxdr 等人，2009)。数据集有 10717 个示例，每个示例都有一对名词性词(无类型)，手工标记为 9 个直接关系中的一个，如产品-生产者(product-producer)(工厂生产套装)或组件-整体(component-whole)(我的公寓有一个大厨房)。

17.2. 关系提取算法

关系提取算法主要有五类：手写模式，有监督的机器学习，半监督(通过自举和远程监督)以及无监督。我们将在接下来的部分中介绍每一个。

17.2.1. 使用模式提取关系

最早也是最常见的关系提取算法是词汇-句法模式，最先由 Hearst (1992a)开发，因此通常称为 **Hearst 模式**。考虑下面的句子:

> Agar is a substance prepared from a mixture of red algae, such as Gelidium, for laboratory or industrial use.

Hearst 指出，大多数人类读者不会知道什么是 Gelidium，但是无论它们是什么，他们都可以轻易地推断出 Gelidium 是一种红藻(的下位词(hyponym))。她建议以下词汇-句法模式：

$$NP_0 \text{ such as } NP_1\{, NP_2 : : : , (and|or)NP_i\}, i \geq 1 \quad (17.2)$$

蕴涵了以下语义：

$$\forall NP_i, i \geq 1, hyponym(NP_i, NP_0) \quad (17.3)$$

让我们推断：

$$\text{hyponym(Gelidium, red algae)} \quad (17.4)$$

NP {, NP}* {,} (and|or) other NP_H	temples, treasuries, and other important civic buildings
NP_H such as {NP,}* {(or|and)} NP	red algae such as Gelidium
such NP_H as {NP,}* {(or|and)} NP	such authors as Herrick, Goldsmith, and Shakespeare
NP_H{,} including {NP,}* {(or|and)} NP	common-law countries, including Canada and England
NP_H{,} especially {NP}* {(or|and)} NP	European countries, especially France, England, and Spain

图 17-5：用于找到上位词的词汇-句法模式

图注：使用{}标记可选性的手工建立的用于找到上位词的词汇-句法模式(Hearst 1992a，Hearst 1998)。

图 17.5 显示了 Hearst (1992a, 1998)提出的推断下位词关系的五种模式;我们已经展示了 NPH 作为父/下位词。基于模式的方法的现代版本通过添加命名实体约束来扩展它。例如，如果我们的目标是回答“谁在哪个组织担任什么职务?”，我们可以使用下面的模式:

PER, POSITION of ORG:
George Marshall, Secretary of State of the United States

PER (named|appointed|chose|etc.) PER Prep? POSITION
Truman appointed Marshall Secretary of State

PER [be]? (named|appointed|etc.) Prep? ORG POSITION
George Marshall was named US Secretary of State

手工构建的模式具有精度高的优点，可以针对特定的领域进行定制。另一方面，它们的召回率低，而且要为所有可能的模式创建它们需要大量的工作。

17.2.2. 通过监督学习提取关系

有监督的机器学习方法来提取关系遵循一个现在应该很熟悉的方案。选择一组固定的关系和实体，用这些关系和实体手工标注训练语料库，然后使用标注的文本训练分类器来标注看不见的测试集。

图 17.6 所示的最简单的方法是:(1)查找成对的命名实体(通常在同一个句子中)。(2)对每一对应用关系分类。分类器可以使用任何监督技术(逻辑回归，RNN，Transformer，随机森林等)。

可选的中间过滤分类器可以通过对给定的命名实体是否相关(通过任何关系)进行二元决策来加快处理速度。它的训练对象是直接从标注的语料库中的所有关系中提取的正示例，以及从没有标注关系的句子内实体对中生成的负示例。

```
function FINDRELATIONS(words) returns relations
  relations ← nil
  entities ← FINDENTITIES(words)
  forall entity pairs <e1, e2> in entities do
    if RELATED?(e1, e2)
        relations← relations+CLASSIFYRELATION(e1, e2)
```

图 17-6：查找并分类文本中实体之间的关系

基于特征的监督关系分类器。让我们考虑基于特征的分类器的样本特征(如逻辑回归或随机森林)，从这句话中对 American Airlines(提及 1，或 M1)和 Tim Wagner(提及 2，或 M2)之间的关系进行分类:

(17.5) **American Airlines**, a unit of AMR, immediately matched the move, spokesman **Tim Wagner** said

这些包括单词特征(如嵌入，或独热，无论词干与否):

•M1 和 M2 的中心词及其连接:

Airlines Wagner Airlines-Wagner

•M1 和 M2 中的词袋和 bigrams:

American, Airlines, Tim, Wagner, American Airlines, Tim Wagner

•特定位置的单词或 bigrams:

M2: -1 spokesman

M2: +1 said

•在 M1 和 M2 之间的词袋或 bigrams:

a, AMR, of, immediately, matched, move, spokesman, the, unit

命名实体特征：

•命名实体类型及其连接：

(M1: ORG, M2: PER, M1M2: ORG-PER)

•M1 和 M2 的实体级别(来自集合 NAME, NOMINAL, PRONOUN)：

M1: NAME [it or he would be PRONOUN]

M2: NAME [the company would be NOMINAL]

•论元之间的实体数(在本例中，AMR 为 1)：

句法结构是一个有用的信号，通常表示为实体之间的树中遍历的依存或成分语法路径。

•M1 和 M2 之间的成分路径：

NP ↑ NP ↑ S ↑ S ↓ NP

•依存树路径：

Airlines $\leftarrow_{subj}$ matched $\leftarrow_{comp}$ said $\rightarrow_{subj}$ Wagner

神经监督关系分类器 用于关系提取的神经模型同样将任务视为监督分类。让我们考虑一个应用于 TACRED 关系提取数据集和任务的典型系统(Zhang 等人，2017)。在 TACRED 中，我们会得到一个句子和其中的两个区段(span)：一个主语，指的是一个人或组织；一个宾语，指的是任何其他实体。任务是从 42 个 TAC 关系中分配一个关系，或者没有关系。

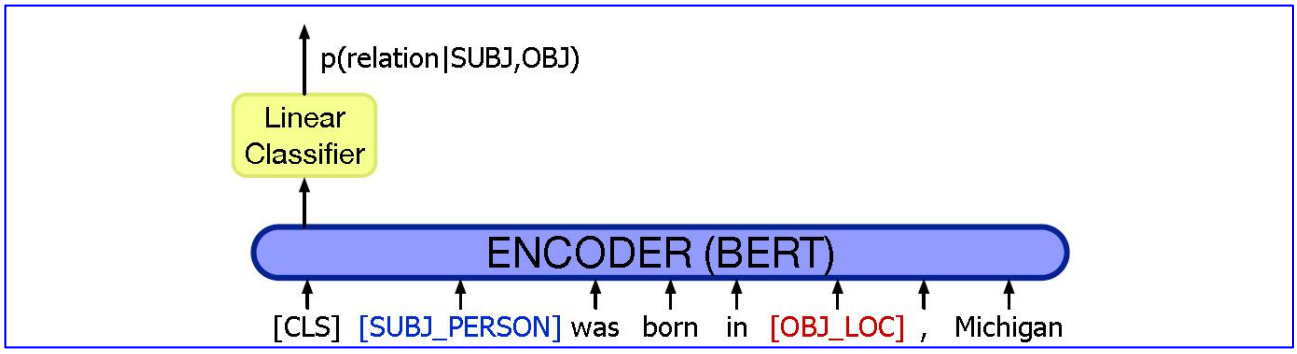

图 17-7：关系提取作为编码器上的线性层(本例中为 BERT)

图注：将输入中的 subject 和 object 实体替换为其 NER 标签(Zhang 等人 2017，Joshi 等人 2020)。

典型的 Transformer-encoder 算法(如图 17.7 所示)仅采用像 BERT 这样的经过预训练的编码器，并在句子表示的顶部添加一个线性层(例如 BERT [CLS]符记)，该线性层被微调为 1-of-N 分类器分配 43 个标签之一。BERT 编码器的输入被部分去混音(de-lexified)。在输入中，主语和宾语实体由其 NER 标记替换。这有助于防止系统过拟合单个词汇项目(Zhang 等人，2017)。当使用 BERT 型 Transformers 进行关系提取时，它有助于使用 RoBERTa (Liu 等人，2019)或 SPANbert (Joshi 等人，2020)等 BERT 版本，这些版本不包含由[SEP]符记分隔的两个序列，而是从单个长句序列中形成输入。

一般来说，如果测试集与训练集足够相似，并且有足够的手工标签数据，监督关系提取系统可获得较高的准确性。但是，标记一个庞大的训练集是非常昂贵的，监督模型是脆弱的：它们不能很好地泛化到不同的文本类型。由于这个原因，关系提取的很多研究都集中在我们接下来要讨论的半监督和非监督方法上。

17.2.3. 通过自举半监督提取关系

监督式机器学习假设我们有很多标签数据。不幸的是，这很贵。但是，假设我们只有一些高精度的**种子模式(seed patterns)**(如第 17.2.1 节中所述)，或者一些**种子元组(seed tuples)**。这足以**自举(bootstrap)**一个分类器!自举技术通过获取种子对中的实体，然后查找包含这两个实体的句子(在 web 上，或我们使用的任何数据集上)。从所有这些句子中，我们提取和概括实体周围的上下文，以学习新的模式。图 17.8 勾画了一个基本算法。

例如，假设我们需要创建一个航空公司/中转站(hub)配对的列表，并且我们只知道 Ryanair 在 Charleroi 有一个中转站。我们可以利用这个种子事实来发现新的模式，通过在语料库中发现其他提及这个关系的情况。我们搜索 Ryanair、Charleroi 和附近的 hub。也许我们能找到下面的句子:

(17.6) Budget airline Ryanair, which uses Charleroi as a hub, scrapped all weekend flights out of the airport.

(17.7) All flights in and out of Ryanair's hub at Charleroi airport were grounded on Friday...

(17.8) A spokesman at Charleroi, a main hub for Ryanair, estimated that 8000 passengers had already been affected.

```
function BOOTSTRAP(Relation R) returns new relation tuples
  tuples← Gather a set of seed tuples that have relation R
  iterate
    sentences← find sentences that contain entities in tuples
    patterns← generalize the context between and around entities in sentences
    newpairs← use patterns to grep for more tuples
    newpairs← newpairs with high confidence
    tuples← tuples + newpairs
  return tuples
```

图 17-8：从种子实体对中通过自举来学习关系

从这些结果中，我们可以使用实体提及(mention)之间的单词上下文，在提及一个之前的单词，在提及两个之后的单词，以及两个提及的命名实体类型，也许还有其他特征，来提取如下的一般模式:

从这些结果中，我们可以使用实体提及之间的词的上下文、第一次提及之前的词、第二次提及之后的词、两次提及的命名实体类型以及其他特征，来提取一般模式，例如:

/ [ORG], which uses [LOC] as a hub /

/ [ORG]'s hub at [LOC] /

/ [LOC], a main hub for [ORG] /

这些新模式可以用于搜索其他元组。

引导系统也为新的元组分配**置信度值(confidence values)**，以避免**语义漂移(semantic drift)**。在语义漂移中，一个错误的模式会导致语义漂移错误元组的引入，进而导致问题模式的产生以及所提取关系"漂移"的含义。考虑以下示例：

(17.9) Sydney has a ferry hub at Circular Quay.

如果作为一个正面的例子被接受，这个表达式可能会导致错误地引入元组<Sydney,CircularQuay>。基于此元组的模式可能会将更多错误传播到数据库中。

模式的置信度值基于两个因素之间的平衡：模式相对于当前元组的性能，以及模式在文档集合中产生的匹配项数量上的生产率。更正式地说，给定文档集合 D，当前元组 T 和提议的模式 p，我们需要跟踪两个因素：

•hits(p)：T 在 D 中查找时匹配的 T 中的元组集

•finds(p)：p 在 D 中找到的元组的总数

下面的方程平衡了这些考虑因素(Riloff 和 Jones, 1999)。

$$Conf_{RlogF}(p) = \frac{|hits(p)|}{|finds(p)|} log(|finds(p)|) \tag{17.10}$$

这个度量标准通常被归一化以产生一个概率。

我们可以通过结合所有与 D 中的元组匹配的模式 P'来支持对新元组的置信度的评估(Agichtein 和 Gravano，2000 年)。结合这种证据的一种方法是"noisy-or"。假定给定的元组由 P 中的模式子集支持，每个模式都有自己的置信度，如上所述。在 noise-or 模型中，我们做出两个基本假设。首先，要使一个提议的元组为假，其所有支持模式都必须是错误的；其次，其单个失败的来源都是独立的。如果我们将置信度度量宽松地视为概率，则任何单个模式 p 失败的概率为 1-Conf(p)；元组所有支持模式错误的可能性是其个别失败概率的乘积，留给我们的等式是我们对一个新的元组的置信度。

$$Conf(t) = 1 - \prod_{p \in P'} (1 - Conf(p)) \tag{17.11}$$

为自举过程中的新模式和元组接受设置保守的置信度阈值，有助于防止系统偏离目标关系。

17.2.4. 通过远程监督提取关系

尽管使用关系标签手工标记文本的生产成本很高，但是有一些方法可以找到训练数据的间接来源。**远程监督(distant supervision)**方法(Mintz 等人，2009)结合了自举和监督学习的优点。远程监督使用一个大型数据库来获取大量的种子样本(而不是少量的种子)，从所有这些样本中创建大量的噪声模式特征，然后将它们组合到一个监督分类器中。

例如，假设我们正在学习人们和他们出生城市之间的 place-of-birth 关系。在基于种子的方法中，我们可能只有 5 个例子可以开始。但基于维基百科的数据库，如 DBPedia 或 Freebase，有成千上万的例子来表达许多关系;包括超过 100,000 个 place-of-birth 的例子，(<Edwin Hubble, Marshfield>，<Albert Einstein, Ulm>，等等，)。下一步是在大量的文本上运行命名实体标记，Mintz 等人(2009)使用了维基百科上的 80 万篇文章，并提取出所有有两个与元组匹配的命名实体的句子，如下所示:

...Hubble was born in Marshfield...

...Einstein, born (1879), Ulm...

...Hubble's birthplace in Marshfield...

现在可以从这个数据中提取训练实例，每个相同的元组都有一个训练实例<relation, entity1, entity2>。因此，每一个都有一个训练实例:

<born-in, Edwin Hubble, Marshfield>

<born-in, Albert Einstein, Ulm>

<born-year, Albert Einstein, 1879> 等等。

然后我们可以应用基于特征或神经的分类。对于基于特征的分类，标准监督关系提取特征，如两个提及的命名实体标签，提及之间的词和依存路径，以及相邻的词。每个元组都有从许多训练实例中收集来的特征;单个训练实例诸如(<born-in,Albert Einstein, Ulm>)的特征向量将具有来自许多提及 Einstein 和 Ulm 的不同句子的词汇和语法特征。

因为远程监督有非常大的训练集，它也能够使用非常丰富的特征，这些特征是这些个体特征的结合。因此，我们将提取数以千计的模式，将实体类型与中间的单词或依存路径连接起来，就像以下这些:

PER was born in LOC

PER, born (XXXX), LOC

PER's birthplace in LOC

回到我们正在运行的例子，这个句子:

(17.12) American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said

我们将学习丰富的连接特征，像这样:

M1 = ORG & M2 = PER & nextword= "said" & path= *NP* ↑ *NP* ↑ *S* ↑ *S* ↓ *NP*

结果是一个有监督的分类器，它有大量丰富的特征用于检测关系。由于不是每个测试句子都有一个训练关系，分类器还需要能够将一个示例标记为无关系(no-relation)。该标签的训练方法是随机选择在任何 Freebase 关系中不出现的实体对，为它们提取特征，并为每个这样的元组构建特征向量。最终的算法如图 17.9 所示。

```
function DISTANT SUPERVISION(Database D, Text T) returns relation classifier C
  foreach relation R
    foreach tuple (e1,e2) of entities with relation R in D
      sentences← Sentences in T that contain e1 and e2
      f ← Frequent features in sentences
      observations← observations + new training tuple (e1, e2, f, R)
  C← Train supervised classifier on observations
  return C
```

图 17-9：关系提取的远程监督算法

图注：神经分类器将跳过特征集 f。

远程监督与我们所研究的每一种方法都有优点。像监督分类，远程监督使用一个分类器与许多特征，并监督详细的手工创建的知识。与基于模式的分类器一样，它可对实体之间的关系使用高精度的证据。实际上，远程监控系统学习模式就像早期关系提取器手工构建的模式一样。例如，Snow 等人(2005)的 is-a 或上位词抽取系统使用 WordNet 中成对的上位词/下位词 NP 作为远程监督，然后从大量文本中学习新模式。他们的系统精确地引入了 Hearst（1992a）的 5 种原始模板模式，但是还引入了 70,000 种其他模式，包括以下 4 种：

NP_H like NP	*Many hormones like leptin...*
NP_H called NP	*...using a markup language called XHTML*
NP is a NP_H	Ruby is a programming language...
NP, a NP_H	*IBM, a company with a long...*

同时使用大量功能的能力意味着，与基于种子的系统中模式的迭代扩展不同，它没有语义漂移。像无监督分类一样，它不使用标记的文本训练语料库，因此它对训练语料库中的体裁问题不敏感，并且依赖大量的未标记数据。远程监督还具有以下优势：可以创建不需要元组的训练元组以便与神经分类器一起使用。

远程监控的主要问题是它往往产生低精度的结果，因此目前的研究集中在提高精度的方法。此外，远程监督只能帮助提取已经存在足够大的数据库的关系。要提取没有数据集的新关系或新领域的关系，必须使用纯无监督的方法。

17.2.5. 通过无监督提取关系

无监督关系提取的目的是在没有标签的训练数据，甚至没有任何关系列表的情况下，从网络中提取关系。这个任务通常被称为**开放信息抽取(open information extraction，Open IE)**。在 Open IE 中，关系只是简单的单词串(通常以动词开头)。例如，ReVerb 系统(Fader 等人，2011)从一个句子 s 中提取关系要通过 4 个步骤:

1. 在 s 上运行一个词类标记器和实体分块器。
2. 对于 s 中的每个动词，找到以一个动词开始并满足语法和词汇约束的单词 w 的最长序列，合并相邻的匹配项。
3. 对于每个短语 w，找到最左边的名词短语 x，它不是关系代词，wh-词或存在的词“there”。在右侧找到最接近的名词短语 y。
4. 使用一个置信度分类器将置信度 c 赋给关系 r = (x,w, y)并返回它。

关系只有在满足句法和词汇限制的情况下才被接受。句法约束确保它是一个以动词开头的序列，也可能包含名词(以轻动词如 make、have 或 do 开头的关系通常表达与名词关系的核心，如 have a hub in):

V | VP | VW*P
V = verb particle? adv?
W = (noun | adj | adv | pron | det)
P = (prep | particle | inf. marker)

词汇约束基于字典 D，该字典用于修剪非常罕见的长关系字符串。直觉是要消除候选关系，这些关系在足够数量的不同论元类型下不会出现，因此可能是不好的例子。该系统首先在 5 亿条网络句子上脱机运行上述关系抽取算法，并抽取出规范化后出现的所有关系列表(去除词形变化、助动词、形容词和副词)。如果每个关系 r 包含至少 20 个不同的论元，则将其添加到字典中。Fader 等人(2011)使用了 170 万个规范化关系的字典。

最后，使用逻辑回归分类器计算每个关系的置信度值。训练分类器的方法是:选取 1000 个随机的 web 句子，运行抽取器，并手工标签每个抽取的关系是正确的还是错误的。然后，使用关系和周围单词的特征，对手工标签的数据训练置信度分类器。图 17.10 显示了一些用于分类的样本特征。

例如下面的句子：

(17.13) United has a hub in Chicago, which is the headquarters of United Continental Holdings.

有关系短语“has a hub in”和“is the headquarters of”(它也有 has 和 is，但更长的短语优先)。步骤 3 在“has a hub in”的左边发现“United”，右边发现“Chicago”，跳过“which”，在“is the headquarters of”的左边发现“Chicago”。最终输出为:

r1: <United, has a hub in, Chicago>

r2: <Chicago, is the headquarters of, United Continental Holdings>

无监督关系抽取的最大优点是它能够处理大量的关系，而不需要预先指定它们。缺点是需要将这些大型字符串集合映射为某种规范形式，以便添加到数据库或其他知识源。目前的方法过于注重用动词表达的关系，因此会遗漏很多**名词性词(nominally)**表达的关系。

(x,r,y) covers all words in s
the last preposition in r is *for*
the last preposition in r is *on*
len(s)≤10
there is a coordinating conjunction to the left of r in s
r matches a lone V in the syntactic constraints
there is preposition to the left of x in s
there is an NP to the right of y in s

图 17-10：分类器的功能

图注：这些功能为开放信息提取系统 REVERB 提取的关系分配可信度(Fader 等人，2011)。

17.2.6. 关系抽取的评价

通过使用带有人类注释的、黄金标准关系的测试集以及计算精度、召回率和 F-measure 来评估**监督**关系抽取系统。标签的精度和召回率要求系统正确地对关系进行分类，而未标签的方法只是测量系统检测相关实体的能力。

半监督(semi-supervised)和**无监督(unsupervised)**方法更难评估，因为它们从 web 或大型文本中提取了全新的关系。因为这些方法使用了大量的文本，通常不可能只在一个小的带标签的测试集中运行它们，因此不可能预先注释一组正确的关系实例。

对于这些方法，可以通过从输出中抽取关系的随机样本来近似(仅)精度，并让人检查每个关系的准确性。通常这种方法侧重于从正文中提取的元组，而不是提及的关系；系统不需要检测每一个提及的关系来得到正确的分数。相反，当系统完成时，评估是基于占用(occupying)数据库的元组集。也就是说，我们想知道系统是否能发现 Ryanair has a hub at Charleroi，我们并不在乎它发现了多少次，则估计精度为：

$$\hat{P} = \frac{\#\text{ of correctly extracted relation tuples in the sample}}{\text{total }\#\text{ of extracted relation tuples in the sample.}} \tag{17.14}$$

另一种方法给了我们一些关于召回率的信息是计算不同召回率级别的精度。假设我们的系统能够对它产生的关系进行排序(根据概率或置信度)，我们可以分别计算前 1000 个新关系、前 10000 个新关系、前 100,000 个新关系的精度，等等。在每一种情况下，我们从该集合中随机抽取样本。这将向我们展示当我们提取越来越多的元组时，精度曲线的表现。但是没有办法直接评估召回率。

17.3. 提取时间

时间和日期是一种特别重要的命名实体，它们在日历和个人助理应用程序的问题回答中扮演着重要角色。为了推断时间和日期，在提取这些**时间表达式(temporal expression)**之后，必须将它们**规范化(normalized)**--转换为标准格式，以便进行推断。在本节中，我们考虑时间表达式的提取和规范化。

17.3.1. 时间表达式提取

时间表达式指的是绝对时间点、相对时间、持续时间和它们的集合。**绝对(absolute)**时间表达式是那些可以直接映射到日历日期、一天中的时间或两者都映射的表达式。**相对(relative)**时间表达式通过一些其他的参考点映射到特定的时间(比如 a week from last Tuesday)。最后，**持续(duration)**时间表示以不同粒度级别(秒、分钟、天、周、世纪等)的时间跨度。图 17.11 列出了这些类别中的一些时间表达式示例。

时态表达是一种语法结构，以时间**词汇触发器(lexical triggers)**作为其中心词。词汇触发器可以是名词、专有名词、形容词和副词;完整的时间表达包括它们的短语投射:名词短语、形容词短语和状语短语。图 17.12 提供了一些示例。

Absolute	Relative	Durations
April 24, 1916	yesterday	four hours
The summer of '77	next semester	three weeks
10:15 AM	two weeks from yesterday	six days
The 3rd quarter of 2006	last quarter	the last three quarters

图 17-11：绝对、关系和持续时间表达式的例子

Category	Examples
Noun	morning, noon, night, winter, dusk, dawn
Proper Noun	January, Monday, Ides, Easter, Rosh Hashana, Ramadan, Tet
Adjective	recent, past, annual, former
Adverb	hourly, daily, monthly, yearly

图 17-12：时间词汇触发器示例

让我们来看看 TimeML 注释方案，其中时间表达式用 XML 标记 TIMEX3 和该标记的各种属性进行注释(Pustejovsky 等人 2005, Ferro 等人 2005)。下面的示例演示了这种模式的基本用法(我们将在 17.3.2 节讨论属性)。

> A fare increase initiated <TIMEX3>last week</TIMEX3> by UAL Corp's United Airlines was matched by competitors over <TIMEX3>the weekend</TIMEX3>, marking the second successful fare increase in <TIMEX3>two weeks</TIMEX3>.

时间表达式识别任务包括查找与这种时间表达式对应的所有文本区间的开始和结束。基于规则的时间表达式识别方法使用自动机级联来识别复杂程度不断增加的模式。符记首先被标记为词类，然后根据包含触发器词(例如，February)或类(例如，MONTH)的模式，从以前阶段的结果中识别出越来越大的组块。图 17.13 给出了一个来自基于规则的系统的片段(Verhagen 等人，2005 年)。

```
# yesterday/today/tomorrow
$string =~ s/(($OT+the$CT+\s+)?$OT+day$CT+\s+$OT+(before|after)$CT+\s+)?$OT+$TERelDayExpr$CT+
(\s+$OT+(morning|afternoon|evening|night)$CT+)?)/<TIMEX$tever TYPE=\"DATE\">$1
<\/TIMEX$tever>/gio;

$string =~ s/($OT+\w+$CT+\s+)<TIMEX$tever TYPE=\"DATE\"[^>]*>($OT+(Today|Tonight)$CT+)
<\/TIMEX$tever>/$1$4/gso;

# this (morning/afternoon/evening)
$string =~ s/(($OT+(early|late)$CT+\s+)?$OT+this$CT+\s*$OT+(morning|afternoon|evening)$CT+)/
<TIMEX$tever TYPE=\"DATE\">$1<\/TIMEX$tever>/gosi;
$string =~ s/(($OT+(early|late)$CT+\s+)?$OT+last$CT+\s*$OT+night$CT+)/<TIMEX$tever
TYPE=\"DATE\">$1<\/TIMEX$tever>/gsio;
```

图 17-13：来自 Tarsqi 中的 GUTime 时间标记系统的 Perl 片段

序列标记方法(Sequence-labeling approaches)遵循与命名实体标记相同的 IOB 方案，用 I、O 和 B 标记在 TIME3 分隔的时间表达式内部、外部或开头的单词，如下所示:

A fare increase initiated last week by UAL Corp's...

O O O O B I O O O

从符记及其上下文中提取特征，并训练统计序列标签器(可以使用任何序列模型)。图 17.14 列出了时间标记中使用的标准特性。

时间表达式识别器是用通常的召回率、精度和 F-测度来评估的。所有这些词汇化的方法的一个主要困难是避免触发误报(false positives，假阳性)的表达:

(17.15) *1984* tells the story of Winston Smith...

(17.16) ...U2's classic *Sunday* Bloody *Sunday*

Feature	Explanation
Token	The target token to be labeled
Tokens in window	Bag of tokens in the window around a target
Shape	Character shape features
POS	Parts of speech of target and window words
Chunk tags	Base phrase chunk tag for target and words in a window
Lexical triggers	Presence in a list of temporal terms

图 17-14：用于训练 IOB 风格时间表达式标记器的典型特征

17.3.2. 时间规范化

时间规范化(temporal normalization)是将时间表达式映射到特定时间点或持续时间的过程。时间点与日历日期、一天中的时间或两者都对应。持续时间主要由时间长度组成，但也可能包括关于开始点和结束点的信息。标准化时间用来自 ISO8601 编码时间值的 VALUE 属性表示(ISO8601, 2004)。图 17.15 复制了我们前面添加了 VALUE 属性的示例 4

```
<TIMEX3 id = ' ' t1 ' ' type ="DATE" value =" 2007-07-02 " functionInDocument ="CREATION TIME"
> July 2, 2007 </TIMEX3> A fare increase initiated <TIMEX3 id =" t2 " type ="DATE"
value =" 2007-W26" anchorTimeID=" t1 ">last week</TIMEX3> by United Airlines was
matched by competitors over <TIMEX3 id =" t3 " type ="DURATION" value ="P1WE"
anchorTimeID=" t1 "> the weekend </TIMEX3>, marking these cond successful fare
increase in <TIMEX3 id =" t4 " type ="DURATION" value ="P2W" anchorTimeID=" t1 "> two
weeks </TIMEX3>.
```

图 17-15：包含时间表达式的规范化值的 TimeML 标注

此文本的日期栏(或文档日期)是 *July 2, 2007*。这种表达式的 ISO 表示形式为 YYYY-MM-DD，在本例中为 2007-07-02。我们的示例文本中时间表达式的编码都是从这个日期开始的，并在此显示为 VALUE 属性的值。

正文中第一个时间表达式是指一年中某个特定的周。在 ISO 标准中，周的编号从 01 到 53，每年的第一个周是当年第一个星期四的那个周。这些星期用模版 YYYY-Wnn 表示。我们文件日期的 ISO 周是第 27 周;因此 last week 的值表示为"2007-W26"。

下一个时间表达式是周末(the weekend)。ISO 的周从星期一开始;因此，周末出现在一周的末尾，并且完全包含在一个星期内。周末被视为持续时间，因此 VALUE 属性的值必须是一个长度。持续时间是根据 Pnx 模式来表示的，其中 n 是一个表示长度的整数，x 表示单位，如 P3Y 表示三年，P2D 表示两天。在本例中，一个周末被捕获为 P1WE。在这种情况下，也有足够的信息来锚定这个特定的周末作为特定一周的一部分。这些信息编码在 ANCHORTIMEID 属性中。最后，短语 two weeks 也表示 P2W 表示的持续时间。关于各种时间注释标准还有很多内容——这里涉及的太多了。图 17.16 描述了表示其他时间和持续时间的一些基本方法。请参阅 ISO8601(2004)、Ferro 等人(2005)和 Pustejovsky 等人(2005)了解更多细节。

目前大多数时间规范化方法都是基于规则的(Chang 和 Manning 2012, Strotgen 和 Gertz 2013)。匹配时间表达式的模式与语义分析过程相关联。就像在第 16 章中介绍的规则到规则的组合方法一样，一个成分的含义是通过使用特定于该成分的方法从其各部分的含义计算出来的，尽管这里的语义组合规则涉及的是时间算法，而不是λ-微积分附件。

Unit	Pattern	Sample Value
Fully specified dates	YYYY-MM-DD	1991-09-28
Weeks	YYYY-Wnn	2007-W27
Weekends	PnWE	P1WE
24-hour clock times	HH:MM:SS	11:13:45
Dates and times	YYYY-MM-DDTHH:MM:SS	1991-09-28T11:00:00
Financial quarters	Qn	1999-Q3

图 17-16：表示不同时间和持续时间的 ISO 模式示例

完全确定(fully qualified)的日期表达式以某些常规形式包含年、月和日。表达式中的单位必须被检测到，然后放在相应 ISO 模式中的正确位置。下面的模式规范了像 April 24, 1916 这样的表达。

FQTE → Month Date , Year {Year.val -- Month.val -- Date.val}

非结束符(non-terminals)月、日和年表示已经被识别和分配语义值的成分，通过*.val 标号访问这些成分。这个 FQE 成分的值，在进一步处理期间，可以反过来通过 FQTE.val 访问 。

在真实文本中，完全确定的时间表达式非常罕见。新闻文章中的大多数时间表达式都是不完整的，并且通常是相对于文章的日期栏而言，是隐式锚定的，我们将其称为文档的**时间锚定(temporal anchor)**。可以相对于该时间锚来计算诸如 today, yesterday, 或者 tomorrow 之类的时间表达式的值。Today 的语义过程仅分配锚点，tomorrow 和 yesterday 的附着词分别从锚点增加一天和减去一天。当然，考虑到我们表示的月份、星期、天和一天中的时间的周期性，我们的时间运算过程必须使用适合所用时间单位的模(modulo)运算。

不幸的是，即使是像 the weekend 或 Wednesday 这样的简单表达式也会引入相当多的复杂性。在我们当前的例子中，the weekend 显然指的是文档日期之前一周的周末。但情况并非总是如此，如下面的例子所示。

(17.17) Random security checks that began yesterday at Sky Harbor will continue at least through the weekend.

在这种情况下，短语 the weekend 指的是锚定日期所在一周中的那个周末(即下一个周末)。表明这个意思的信息来自于 continue 时态，这个动词支配着 the weekend。

相对时间表达式使用时间算法处理，类似于 today 和 yesterday 使用的算法。文档日期表明我们的示例文章是 ISO week 27，因此 last week 的表达式规范化为当前周减 1。为了解决有歧义的 next 和 last 表达式，我们考虑从锚定日期到最近单位的距离。Next Friday 可以指的是靠近当前的下一个星期五，也可以指下一周的星期五，但是文档日期离星期五越近，则该短语越有可能跳过最接近的那个星期五。通过将语言和特定于领域的试探法编码到时间附着词中，可以处理此类歧义。

17.4. 提取事件及其时间

事件提取(event extraction)的任务是识别文本中提到的事件。在本任务中，提及的事件是任何表示事件或状态的表达式，该事件或状态可以在时间上分配给特定的点或间隔。下面示例文本的标记显示了该文本中的所有事件。

[EVENT Citing] high fuel prices, United Airlines [EVENT said] Friday it has [EVENT increased] fares by $6 per round trip on flights to some cities also served by lower-cost carriers. American Airlines, a unit of AMR Corp., immediately [EVENT matched] [EVENT the move], spokesman Tim Wagner [EVENT said]. United, a unit of UAL Corp., [EVENT said] [EVENT the increase] took effect Thursday and [EVENT applies] to most routes where it [EVENT competes] against discount carriers, such as Chicago to Dallas and Denver to San Francisco.

在英语中，大部分事件都对应于动词，而动词大多是用来介绍事件的。然而，从我们的例子中可以看出，情况并非总是如此。事件可以由名词短语引入，如在 the move 和 the increase 中，而有些动词不能引入事件，如动词短语 took effect，它指的是事件开始的时间，而不是事件本身。同样，轻动词，如 make、take 和 have 也常常不能表示事件;对于轻动词来说，事件通常用名词性直接宾语(took a flight)来表示，这

些轻动词只是为名词的论元提供了一个句法结构。

根据目标的不同，存在不同版本的事件提取任务。例如，在 TempEval 共享任务(Verhagen 等人，2009)中，目标是提取事件和体(aspects)，比如它们的体和时间属性。事件被分类为动作、状态、**报告事件(reporting events)**(比如 say、report、tell、explain)、感知事件，等等。每个事件的体、时态和情态也需要提取出来。例如，示例文本中的各种 said 事件可以注释为(class=REPORTING, tense=PAST, aspect=PERFECTIVE)。

事件提取通常通过监督学习建模，通过带有 IOB 标记的序列模型检测事件，并使用多类分类器分配事件类和属性。基于特征的模型使用表面信息，如词类、词汇项和动词时态信息;见图 17.17。

Feature	Explanation
Character affixes	Character-level prefixes and suffixes of target word
Nominalization suffix	Character-level suffixes for nominalizations (e.g., -tion)
Part of speech	Part of speech of the target word
Light verb	Binary feature indicating that the target is governed by a light verb
Subject syntactic category	Syntactic category of the subject of the sentence
Morphological stem	Stemmed version of the target word
Verb root	Root form of the verb basis for a nominalization
WordNet hypernyms	Hypernym set for the target

图 17-17：在事件检测中基于规则和机器学习方法的常用特征

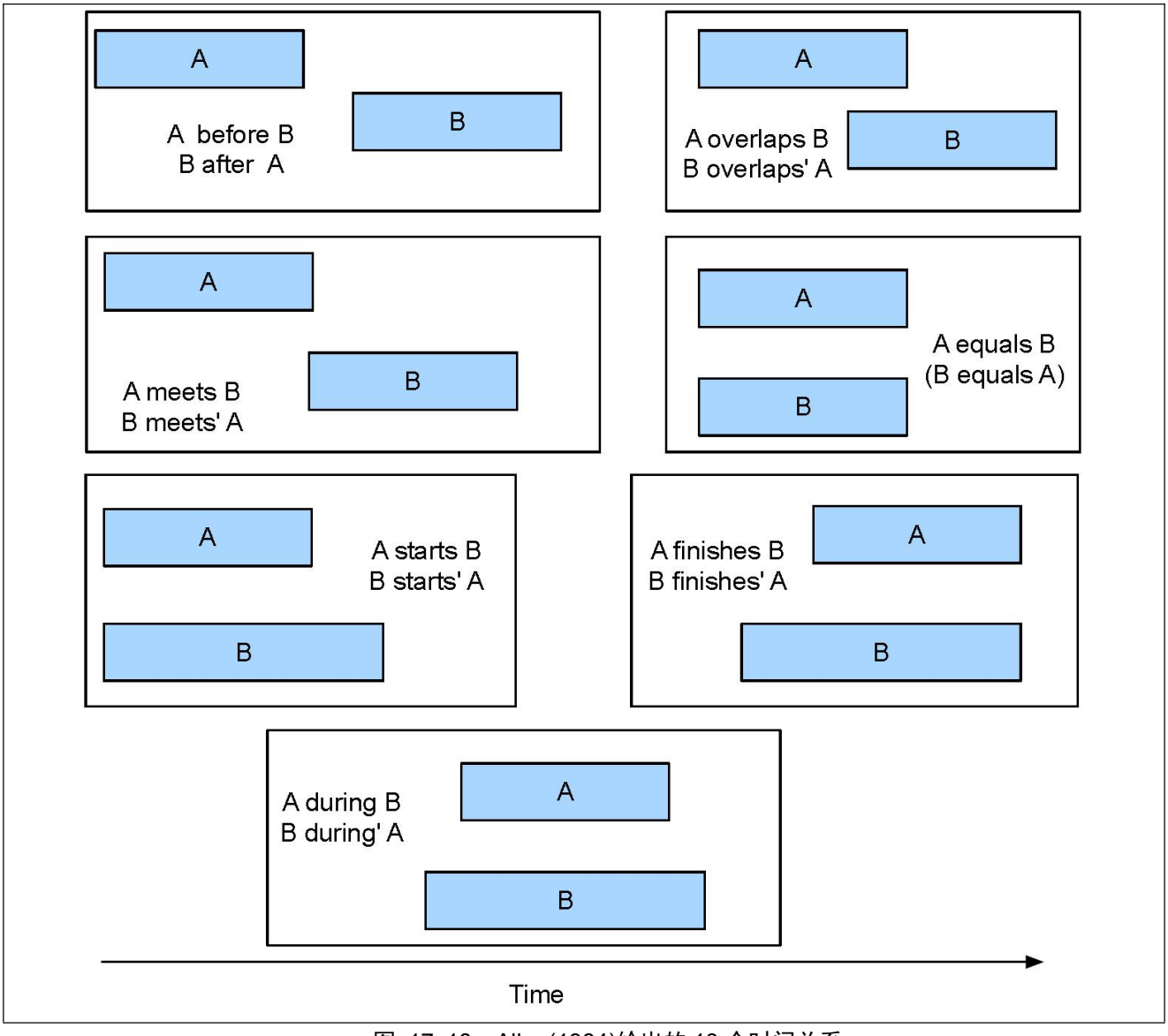

图 17-18：Allen(1984)给出的 13 个时间关系

17.4.1. 事件的时间顺序

在检测到文本中的事件和时态表达式之后，下一个逻辑任务是使用这些信息将事件整合到一个完整的时间轴中。这样的时间表对诸如问答和摘要等应用程序是有用的。这项雄心勃勃的任务是目前大量研究的主题，但超出了当前系统的能力。

一个稍微简单但仍然有用的任务是对文本中提及的事件和时间表达式进行部分排序。这样的排序可以提供与真正的时间表相同的许多好处。这种部分排序的一个示例是，在我们的示例文本中确定美国航空公司的票价上涨是在联合航空公司的票价上涨之后。确定这样的排序可以看作是一种二元关系检测和分类任务，类似于前面第 17.1 节中描述的任务。事件之间的时间关系被分类为图 17.18 (Allen, 1984)所示的 **Allen 关系(Allen relations)**标准集之一，使用第 17.1 节所示的基于特征的分类器，在 TimeBank 语料库上进行训练，特征包括单词/嵌入、解析路径、时态和体(aspect)。

时间库(TimeBank)语料库由注释了我们在本节中讨论过的许多信息的文本组成(Pustejovsky 等人，2003b)。时间库 1.2 包含从各种来源(包括宾州树库和 PropBank 集合)选择的 183 个新闻文章。

时间库语料中的每一篇文章都有在 TimeML 注释中明确标注的时间表达式和事件(Pustejovsky 等人，2003a)。除了时间表达式和事件之外，TimeML 注释还提供了事件和时间表达式之间的时间连接，后者指定了它们之间关系的性质。考虑下面的示例句子和图 17.19 中所示的相应标记，它们是从一个 TimeBank 文档中选择的。

(17.18) Delta Air Lines earnings soared 33% to a record in the fiscal first quarter, bucking the industry trend toward declining profits.

如注释所示，该文本包括三个事件和两个时间表达式。这些事件都在发生(occurrence)类中，并被赋予了唯一的标识符，以便在进一步的注释中使用。时间表达式包括文章的创建时间(作为文档时间)和文本中的单个时间表达式。

除了这些注释之外，TimeBank 还提供了四个连接，使用图 17.18 中的 Allen 关系来捕捉文本中事件和时间之间的时间关系。下面是为这个例子注释的句子内时间关系。

- Soaring$_{e1}$ is **included** in the fiscal first quarter$_{t58}$
- Soaring$_{e2}$ is **before** 1989-10-26$_{t57}$
- Soaring$_{e3}$ is **simultaneous** with the bucking$_{e3}$
- Declining$_{e4}$ **includes** soaring$_{e1}$

```
<TIMEX3 tid="t57" type="DATE" value="1989-10-26" functionInDocument="CREATION_TIME">
10/26/89 </TIMEX3>
Delta Air Lines earnings <EVENT eid="e1" class="OCCURRENCE"> soared </EVENT> 33% to a
record in <TIMEX3 tid="t58" type="DATE" value="1989-Q1" anchorTimeID="t57"> the
fiscal first quarter </TIMEX3>, <EVENT eid="e3" class="OCCURRENCE">bucking</EVENT>
the industry trend toward <EVENT eid="e4" class="OCCURRENCE">declining</EVENT>
profits.
```

图 17-19：来自 TimeBank 语料库的示例

17.5. 模板填充

许多文本包含对事件的报道，以及可能的事件序列，这些报道往往与世界上相当常见的、刻板的情况相对应。这些抽象的情境或故事，与所谓的**脚本(scripts)**(Schank 和 Abelson, 1977)有关，由子事件、参与者和他们的角色的原型序列组成。这些脚本提供的强烈期望可以促进实体的适当分类，将实体分配到角色和关系中，以及最重要的是，对填充未说出的内容的推论的描绘。在最简单的形式中，这类脚本可以表示为由固定的槽集组成的模板，这些槽集接受属于特定类的**槽填充符(slot-fillers)**作为值。**模板填充(template filling)**的任务是找到调用特定脚本的文档，然后用从文本中提取的填充符来填充相关模板中的槽。这些槽填充符可以由直接从文本中提取的文本段组成，也可以由通过一些额外处理从文本元素中推断

出来的概念组成。

我们最初的航空公司故事中的填充模板可能如下所示。

```
FARE-RAISE ATTEMPT: [ LEAD AIRLINE:    UNITED AIRLINES
                      AMOUNT:          $6
                      EFFECTIVE DATE:  2006-10-26
                      FOLLOWER:        AMERICAN AIRLINES ]
```

该模板有四个槽(LEAD AIRLINE, AMOUNT, EFFECTIVE DATE, FOLLOWER)。下一节描述填充槽的标准序列标记方法。第 17.5.2 节接着描述了一个基于**有限状态转录机(fifinite-state transducers)**级联使用的旧系统，该系统旨在解决当前基于学习的系统尚未解决的更复杂的模板填充任务。

17.5.1. 模板填充的机器学习方法

在模板填充的标准范例中，我们得到了带有预定义模板及其槽填充符注释的文本跨度的训练文档。我们的目标是为输入中的每个事件创建一个模板，用文本跨度填充槽。

这个任务通常是通过训练两个独立的监督系统来建模的。

第一个系统决定模板是否存在于特定的句子中。这个任务被称为**模板识别(template recognition)**，或者有时，用一个可能令人困惑的术语来说，称为事件识别。模板识别可以被视为文本分类任务，从训练文档中标记的每个单词序列中提取特征，填补被检测的模板中的任何位置。可以使用通常的一组特征：符记、嵌入、单词形状、词类标记、语法块标记和命名实体标记。

第二个系统的工作是**角色填充符提取(role-filler extraction)**。训练一个单独的分类器来检测每个角色(LEAD-AIRLINE、AMOUNT，等等)。它可以是在已解析输入句子中的每个名词短语上运行的二元分类器，也可以是在单词序列上运行的序列模型。每个角色分类器都根据训练集合中的标签数据进行训练。同样，可以使用通常的特征集，但现在只针对单个名词短语或单个槽的填充符进行训练。

多个不相同的文本段可以用相同的槽标签进行标记。例如，在我们的示例文本中，字符串 United 或 United Airlines 可能被标记为 LEAD AIRLINE。这些并不是不兼容的选择，第 22 章中介绍的共指解析技术可以提供一个解决方案。

各种注释集合已经被用来评估这种模板填充方法，包括工作公告集合、论文会议电话、餐厅指南和生物文本。最近的工作侧重于在没有训练数据或甚至没有预定义模板的情况下提取模板，通过将模板归纳为一系列关联事件(Chambers 和 Jurafsky, 2011)。

17.5.2. 早期的有限状态模板填充系统

上面的模板相对简单。但是考虑一下生成一个模板的任务，该模板在这样的文本中包含所有信息(Grishman 和 Sundheim，1995)：

Bridgestone Sports Co. said Friday it has set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be shipped to Japan. The joint venture, Bridgestone Sports Taiwan Co., capitalized at 20 million new Taiwan dollars, will start production in January 1990 with production of 20,000 iron and “metal wood” clubs a month.

MUC-5 的“合资企业”任务(信息理解会议是一系列美国政府组织的信息提取评估)是生成描述合资企业的层次连接模板。图 17.20 显示了 FASTUS 系统产生的结构(Hobbs 等人，1997)。请注意，绑定模板的 ACTIVITY 槽的填充器本身是一个带有槽的模板。

早期处理这些复杂模板的系统是基于手写规则的转录机级联，如图 17.21 所示。

Tie-up-1		Activity-1:	
RELATIONSHIP	tie-up	COMPANY	Bridgestone Sports Taiwan Co.
ENTITIES	Bridgestone Sports Co. a local concern a Japanese trading house	PRODUCT START DATE	iron and “metal wood” clubs DURING: January 1990
JOINT VENTURE	Bridgestone Sports Taiwan Co.		
ACTIVITY	Activity-1		
AMOUNT	NT$20000000		

图 17-20：FASTUS 在输入文本后生成的模板

No.	Step	Description
1	Tokens	Tokenize input stream of characters
2	Complex Words	Multiword phrases, numbers, and proper names.
3	Basic phrases	Segment sentences into noun and verb groups
4	Complex phrases	Identify complex noun groups and verb groups
5	Semantic Patterns	Identify entities and events, insert into templates.
6	Merging	Merge references to the same entity or event

图 17-21：FASTUS 中的处理水平

图注：FASTUS 中的处理水平(Hobbs 等人，1997)。每一层提取特定类型的信息，然后传递到下一层。

前四个阶段使用手写的正则表达式和语法规则进行基本的符记化、分块和解析。然后，第 5 阶段使用基于 FST 的识别器识别实体和事件，并将识别的对象插入模板中适当的槽中。这个 FST 识别器基于如下手工构建的正则表达式(NG 表示名词组，VG 表示动词组)，它匹配上面新闻故事的第一个句子。

NG(Company/ies) VG(Set-up) NG(Joint-Venture) with NG(Company/ies) VG(Produce) NG(Product)

处理这两个句子的结果是五个草案模板(图 17.22)，然后必须合并到图 17.20 所示的单一层次结构中。在执行了相关解析之后，合并算法将合并两个可能描述相同事件的活动。

# Template/Slot	Value
1 RELATIONSHIP:	TIE-UP
ENTITIES:	Bridgestone Co., a local concern, a Japanese trading house
2 ACTIVITY:	PRODUCTION
PRODUCT:	“golf clubs”
3 RELATIONSHIP:	TIE-UP
JOINT VENTURE:	“Bridgestone Sports Taiwan Co.”
AMOUNT:	NT$20000000
4 ACTIVITY:	PRODUCTION
COMPANY:	“Bridgestone Sports Taiwan Co.”
STARTDATE:	DURING: January 1990
5 ACTIVITY:	PRODUCTION
PRODUCT:	“iron and “metal wood” clubs”

图 17-22：FASTUS 的阶段 5 生成的 5 个部分模板

图注：这些模板在阶段 6 中合并，以生成图 17.20 中所示的最终模板。

17.6. 总结

本章探讨了从文本中提取有限形式的语义内容的技术。

•**实体之间的关系**可以通过基于模式的方法提取：当有注释的训练数据可用时，可以使用**监督**学习方法；当有少量的**种子元组**或**种子模式**可用时，可以使用轻度**监督自举**方法；当有关系数据库可用时，可以使用**远程监督**、**无监督**或 **OPEN IE** 方法。

•通过结合统计学习和基于规则的方法对**时间表达式**进行检测和规范化，可以促进对时间的推理。

•使用经过时间和事件标记的数据(如 **TimeBank 语料库**)训练的序列模型和分类器，可以及时检测和排序**事件**。

•**模板填充**应用程序可以识别文本中的定型情况，并将文本中的元素分配给表示为**固定槽集合**的角色。

17.7. 文献和历史说明

最早的信息提取工作是在 Frump 系统的背景下处理模板填充任务(DeJong, 1982)。后来的研究受到了美国政府主办的 MUC 会议的推动(Sundheim 1991、Sundheim 1992、Sundheim 1993、Sundheim 1995)。早期的 MUC 系统，如 CIRCUS 系统 (Lehnert 等人，1991)和 SCISOR (Jacobs 和 Rau, 1990)，对后来的系统诸如 FASTUS (Hobbs 等人，1997)产生了很大的影响和启发。Chinchor 等人(1993)描述了 MUC 评价技术。

由于将系统从一个领域移植到另一个领域很困难，人们的注意力转向了机器学习方法。早期的 IE 监督学习方法(Cardie 1993, Cardie 1994, Riloff 1993, Soderland 等人 1995, Huffman 1996)关注于知识获取过程的自动化，主要用于有限状态的基于规则的系统。他们的成功，以及早期基于 HMM 的语音识别的成功，导致了序列标记的使用(HMMs：Bikel 等人 1997；MEMMs：McCallum 等人 2000；CRFs：Lafferty 等人 2001)，以及对特征的广泛探索(Zhou 等人，2005)。神经方法遵循了 Collobert 等人(2011)的开创性成果，他们在卷积网络上应用了 CRF。

通过使用共享基准数据集进行正式评估，继续刺激该领域的进展，包括 2000-2007 年对命名实体识别，关系提取和时间表达的自动内容提取(ACE)评估，**KBP**(Knowledge Base Population，知识基础群体)评估(Ji 等人 2010，Surdeanu 2013)的关系提取任务，例如槽填充(提取给定实体的年龄，出生地和配偶等属性(“slots”))和一系列 SemEval 研讨会(Hendrickx 等人，2009) 。

半监督关系提取最先由 Hearst (1992b)提出，并由 AutoSlog-TS (Riloff, 1996)、DIPRE (Brin, 1998)、SNOW-BALL (Agichtein 和 Gravano, 2000)和 Jones 等(1999)系统进行了扩展。远程监督算法描述从 Mintz 等人(2009),他们创造了“远程监督”这个词,但类似的想法发生在早期的系统中，该系统冠以“weakly labeled data(弱带安全标签的数据)”的名义(Craven 和 Kumlien(1999)和 Morgan 等人 2004，以及 Snow 等人 2005,Wu 和 Weld 2007)。这些系统还有很多扩展，其中包括 Wu 和 Weld(2010)、Riedel 等人(2010)和 Ritter 等人(2013)。开放 IE 系统包括 KNOWITALL(Etzioni 等人，2005)、TextRunner (Banko 等人，2007)和 REVERB (Fader 等人，2011)。还有一种通用模式，该模式结合了远程监督和 OPEN IE 的优势，参见 Riedel 等人(2013)。

HeidelTime (Strotgen 和 Gertz, 2013)和 SUTime (Chang 和 Manning, 2012)是可下载的时间提取和规范化系统。UzZaman 等人(2013)描述了 2013 年的 TempEval 挑战;Chambers(2013)和 Bethard(2013)给出了典型的方法。

17.8. 练习

17.1 Acronym expansion, the process of associating a phrase with an acronym, can be accomplished by a simple form of relational analysis. Develop a system based on the relation analysis approaches described in this chapter to populate a database of acronym expansions. If you focus on English **Three Letter Acronyms** (TLAs) you can evaluate your system’s performance by comparing it to Wikipedia’s TLA page.

17.2 A useful functionality in newer email and calendar applications is the ability to associate temporal expressions connected with events in email (doctor’s appointments, meeting planning, party invitations, etc.) with specific calendar entries. Collect a corpus of email containing temporal expressions related to event planning. How do these expressions compare to the kinds of expressions commonly found in news text that we’ve been discussing in this chapter?

17.3 Acquire the CMU seminar corpus and develop a template-filling system by using any of the techniques mentioned in Section 17.5. Analyze how well your system performs as compared with SOTA results on this corpus.