知识图谱模式设计

知识图谱模式

知识图谱是实体和实体之间的关系所构成的网络，是人类和机器都能使用的知识表示方法。

对于实体，用键值对<属性名，属性值>的方式从多视角描述实体的不同维度特征，即为实体属性。对于相同类型的实体，属性名相同，即可以用实体类型中的属性名列表来表示一类实体的共同的多维特征。

从实体类型的角度挖掘出知识图谱中的关系，可以推演出以<头实体的实体类型，关系，尾实体的实体类型>来抽象描述一系列的关系三元组，即关系类型。关系也有<属性名，属性值>形式的属性。

将实体类型、实体类型的属性名列表、关系类型、关系类型的属性名列表汇总到一起，就构成了知识图谱的语义化的规范，即知识图谱模式。

知识图谱模式：简称模式，是面向知识图谱内容的一种抽象的、语义化的且概念化的规范。在语义网中，知识图谱模式往往也被称为本体（Ontology），表示知识的概念化的规范。

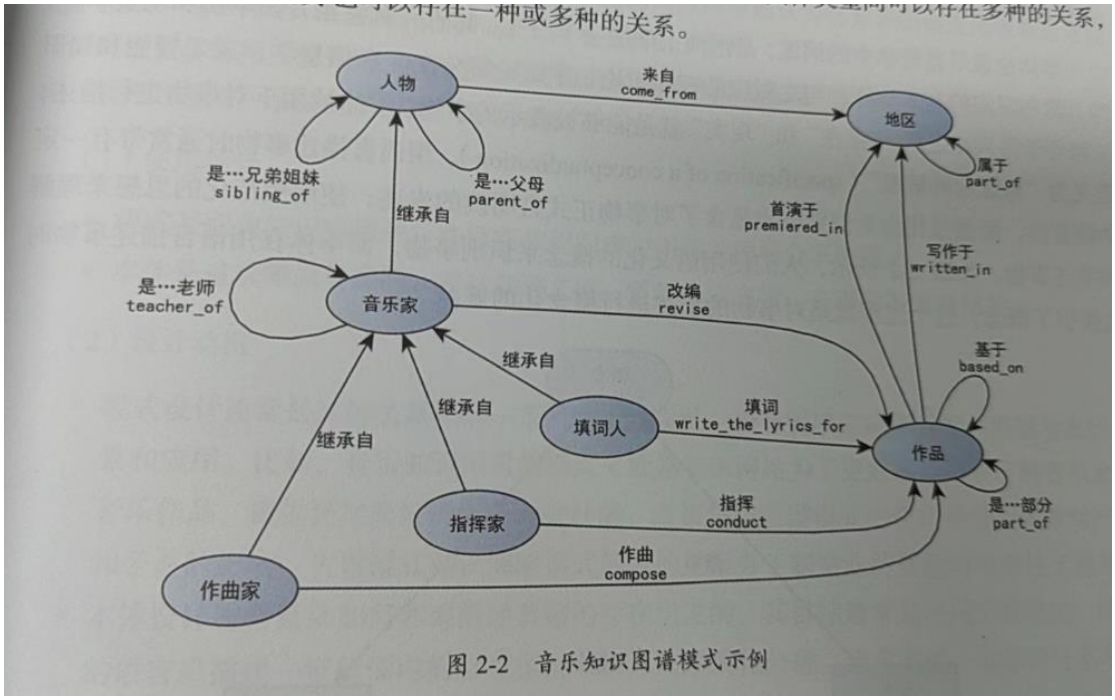


图 2-2　音乐知识图谱模式示例

模式和本体

本体的核心目标是通过定义一组领域内的概念和类别，以及它们之间的关系来组织信息和知识。本体被广泛用在知识图谱领域，与知识图谱模式所表达的概念大同小异。

模式更多偏向于工程实践和应用落地，本地更多追求知识的本质。

本体概论

本体的构成要素：

（1） 实例：是本体的底层对象，类似于知识图谱中的实体，是类或概念实例化的对象。本体通常不包括实例。

（2） 类 class：也称概念或类型，在知识图谱模式中被叫做实体类型，类是对事物进行分组和抽象。本体的基础元素是类。

（3） 属性：类可能具有的属性

（4） 关系：类和类、概念与概念之间可能存在的关联关系。

（5） 规则

（6） 公理

本体的分类：

（1） 基础本体：是对现实世界普遍适用的通识进行建模，其中收录了适用于多个不同领域的共有的或核心的概念和术语。如 Schems、COSMO。

（2） 领域本体：如金融行业业务本体 FIBO、基因本体 GO。如果要设计或创建一个领域本体，那么基础本体是很好的起点，可以从基础本体中提取适合本领域的概念、术语、属性和关系等知识。

资源描述框架 RDF，是一个基础且通用的数据模型，用于表示语义网的资源信息。RDF 模式（RDFS）和网络本体语言（OWL）都建立在 RDF 之上。

模式设计的三个基本原则

模式设计的方法-六韬法

知识图谱构建技术

确定了知识图谱模式后，知识图谱构建技术可以源源不断把数据转换为知识。知识图谱的构建过程就是根据知识来源选择合适的技术，实现从数据到知识的转换，知识来源分为非结构化数据和结构化数据，因此知识图谱构建技术分为映射式构建技术和抽取式构建技术。

结构化数据：对于结构化数据使用映射式构建技术，通常根据结构化数据源和目标知识图谱的要求，设定、配置或编写一系列的规则来实现。

非结构化数据：对于非结构化数据使用抽取式构建技术，核心是从非结构化数据源中提取实体和关系。非结构化数据包括文本、图像和声音，现阶段常见的知识图谱大多是基于文本的，即使在多模态知识图谱中，图像、视频、语音等常用于展示。

实体抽取

命名实体识别（Named Entity Recognition）是指从非结构化的文本中识别出符合定义的实体，并将其分类到某个恰当实体类型中。 在知识图谱领域，广义的命名实体识别通常又称为实体抽取，实体抽取和命名实体识别采用相同的技术。最早的实体抽取方法是基于规则的，由专家编写规则来抽取实体，抽取效果完全依赖于人工编写的规则，具有规则繁琐、泛化能力差、成本高、总体效率低等局限性。然后，基于机器学习和统计学习的方法出现了，这些方法通常采用较为通用的大量特征，致力于让模型从大量的样本中学习出特征的模式（pattern），从

而实现高效和泛化能力强的实体抽取。近年来，基于深度学习的知识抽取方法逐渐成为主流，最显著的特点是不需要人工选择特征，而是让模型从样本数据中同时学习特征和陌生，效果更好、效率更高、泛化能力更强。

为了解决少样本的问题，弱监督学习的实体抽取方法也被广泛研究。

基于规则的实体抽取

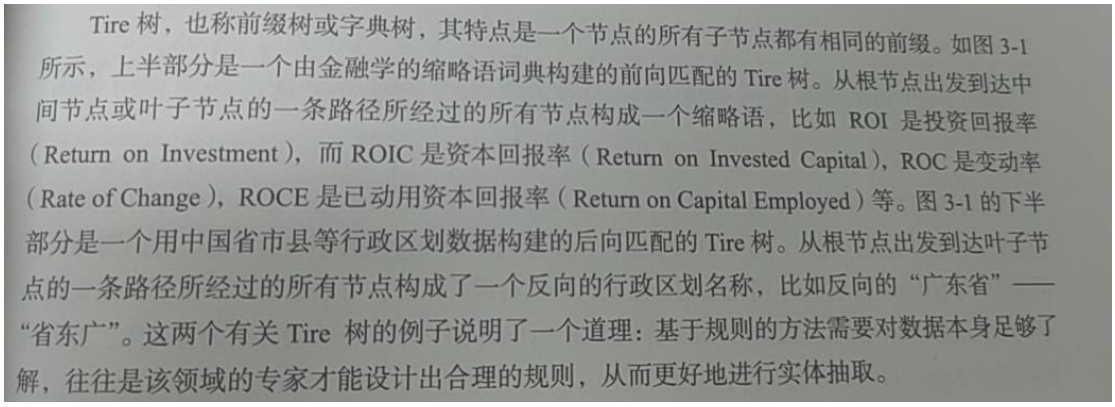
基于规则的方法是最早用来抽取实体的方法，在很多场景下至今也是很好的选择。

（1）基于词典匹配的实体抽取方法

词典匹配是一种最简单的基于规则的实体抽取方法。在具有业务词典的场景下，比如地名词典、各领域专名词典等，通过词典匹配从文本中抽取实体很高效。

词典匹配的第一步是使用 tokenizer（分词器）对文本进行分词，把文本转换为 token 序列，将 token 序列与词典进行匹配，获得实体。

在进行词典匹配时，常用到 Trie 树。



Tire 树，也称前缀树或字典树，其特点是一个节点的所有子节点都有相同的前缀。如图 3-1 所示，上半部分是一个由金融学的缩略语词典构建的前向匹配的 Tire 树。从根节点出发到达中间节点或叶子节点的一条路径所经过的所有节点构成一个缩略语，比如 ROI 是投资回报率（Return on Investment），而 ROIC 是资本回报率（Return on Invested Capital），ROC 是变动率（Rate of Change），ROCE 是已动用资本回报率（Return on Capital Employed）等。图 3-1 的下半部分是一个用中国省市县等行政区划数据构建的后向匹配的 Tire 树。从根节点出发到达叶子节点的一条路径所经过的所有节点构成了一个反向的行政区划名称，比如反向的“广东省”——“省东广”。这两个有关 Tire 树的例子说明了一个道理：基于规则的方法需要对数据本身足够了解，往往是该领域的专家才能设计出合理的规则，从而更好地进行实体抽取。

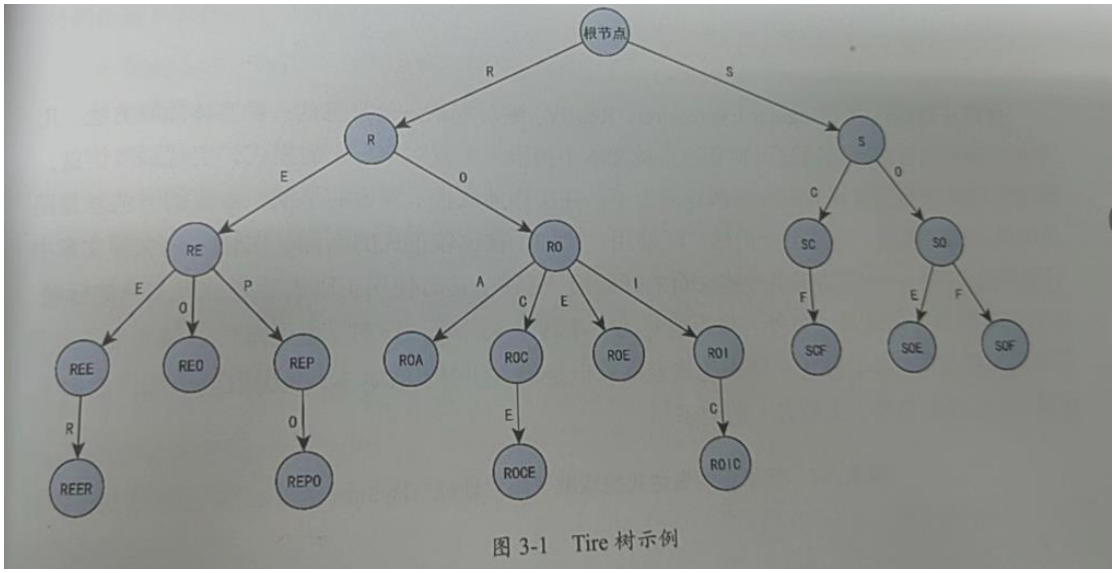


图 3-1 Tire 树示例

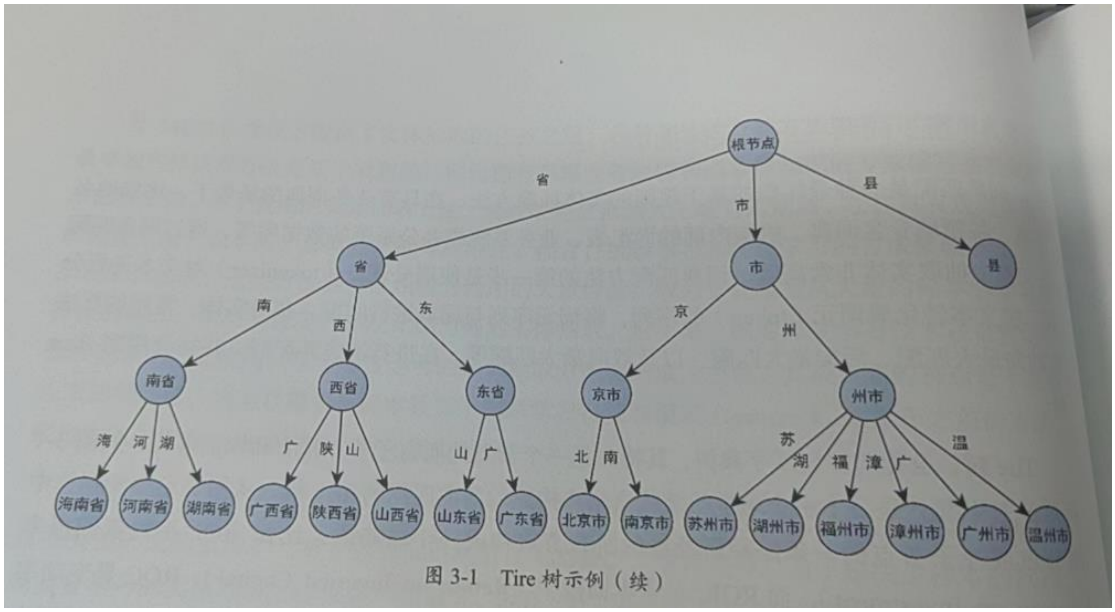


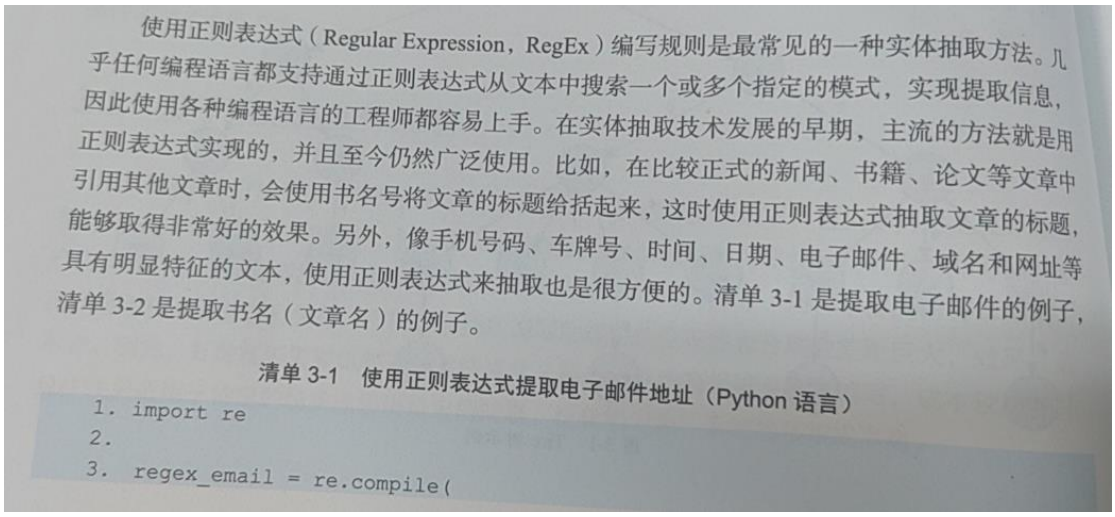
图 3-1 Tire 树示例（续）

大部分的分词库都包含 Trie 树的实现，对于中文，很多分词库都支持词典导入的功能：将词典导入分词库中，利用分词库将实体词完整切分出来，再进行词典匹配，可以更简单实现基于词典匹配的实体抽取。

Jieba 分词库。

（2）编写正则表达式抽取实体

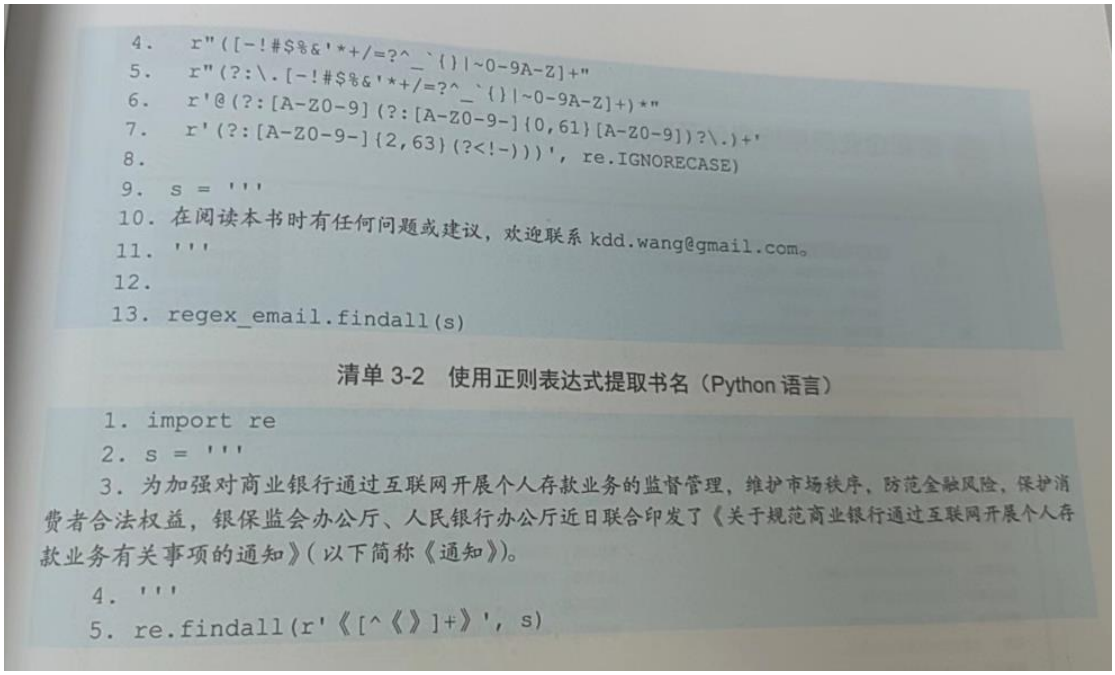
像手机号码、车牌号、时间、日期、电子邮件、域名和网址等都是具有明显特征的文本，使用正则表达式来抽取也是很方便的。



使用正则表达式（Regular Expression，RegEx）编写规则是最常见的一种实体抽取方法。几乎任何编程语言都支持通过正则表达式从文本中搜索一个或多个指定的模式，实现提取信息，因此使用各种编程语言的工程师都容易上手。在实体抽取技术发展的早期，主流的方法就是用正则表达式实现的，并且至今仍然广泛使用。比如，在比较正式的新闻、书籍、论文等文章中引用其他文章时，会使用书名号将文章的标题给括起来，这时使用正则表达式抽取文章的标题，能够取得非常好的效果。另外，像手机号码、车牌号、时间、日期、电子邮件、域名和网址等具有明显特征的文本，使用正则表达式来抽取也是很方便的。清单 3-1 是提取电子邮件的例子，清单 3-2 是提取书名（文章名）的例子。

清单 3-1　使用正则表达式提取电子邮件地址（Python 语言）

```
1. import re
2.
3. regex_email = re.compile(
```



```
4.     r"([-!#$%&'*+/=?^_`{}|~0-9A-Z]+"
5.     r"(?:\.[-!#$%&'*+/=?^_`{}|~0-9A-Z]+)*"
6.     r'@(?:[A-Z0-9](?:[A-Z0-9-]{0,61}[A-Z0-9])?\.)+'
7.     r'(?:[A-Z0-9-]{2,63}(?<!-)))', re.IGNORECASE)
8.
9.  s = '''
10. 在阅读本书时有任何问题或建议，欢迎联系 kdd.wang@gmail.com。
11. '''
12.
13. regex_email.findall(s)
```

清单 3-2　使用正则表达式提取书名（Python 语言）

```
1. import re
2. s = '''
3. 为加强对商业银行通过互联网开展个人存款业务的监督管理，维护市场秩序，防范金融风险，保护消
费者合法权益，银保监会办公厅、人民银行办公厅近日联合印发了《关于规范商业银行通过互联网开展个人存
款业务有关事项的通知》（以下简称《通知》）。
4. '''
5. re.findall(r'《[^《》]+》', s)
```

（3）基于模板的实体抽取方法

基于模板的实体抽取方法通常用在有一定结构的文本（结构化或半结构化文本）中，特别是基于模板自动生成的文档或网页中。

XPath（XML Path Language）是一种定位元素并抽取实体的工具，通常用在网页中。

对这样结构化或半结构化的文本进行实体抽取，基于模板的方法是非常有效的。XPath（XML Path Language）是一种定位元素并抽取实体的工具，通常用在网页中。以“国家企业信用信息公示系统”网站为例，页面如图 3-2 所示，利用简单的程序语句“xpath('//div[@id="primaryInfo"]')”即可获取“营业执照信息”部分的内容（见清单 3-3），然后利用表格模板实现实体抽取，比如法定代表人等。

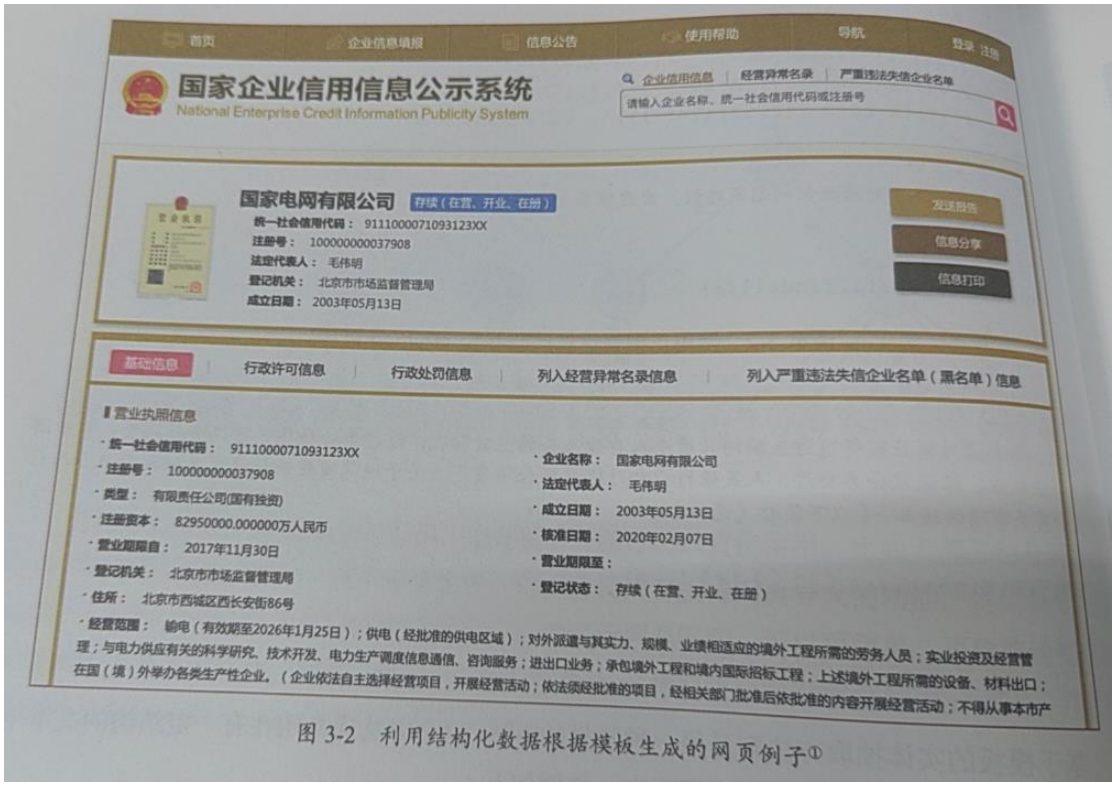


图 3-2 利用结构化数据根据模板生成的网页例子①

清单 3-3 XPath 的输出结果，是一个 html 格式的表格

```
1.  <div id="primaryInfo" class="tabin mainContent">
2.    <div class="details clearfix">
3.      <div class="classify">营业执照信息</div>
4.      <div class="overview">
5.        <dl>
6.          <dt class="item"> 统一社会信用代码: </dt>
7.          <dd class="result"><!--这里还需要添加业务逻辑 -->91110000710931 23XX
</dd>
8.        </dl>
9.        <dl>
10.         <dt class="item_right">企业名称: </dt>
11.         <dd class="result" title="国家电网有限公司">国家电网有限公司</dd>
12.       </dl>
13.       <dl>
14.         <dt class="item"> 注册号: </dt>
15.         <dd class="result"><!-- 这里还需要添加业务逻辑 -->100000000037908</dd>
```

① 内容来自“国家企业信用信息公示系统”，2020-12-28。

Lxml 是一个很成熟的 Python 语言库，可以用来进行 XPath 抽取。

（4）评价实体抽取的效果

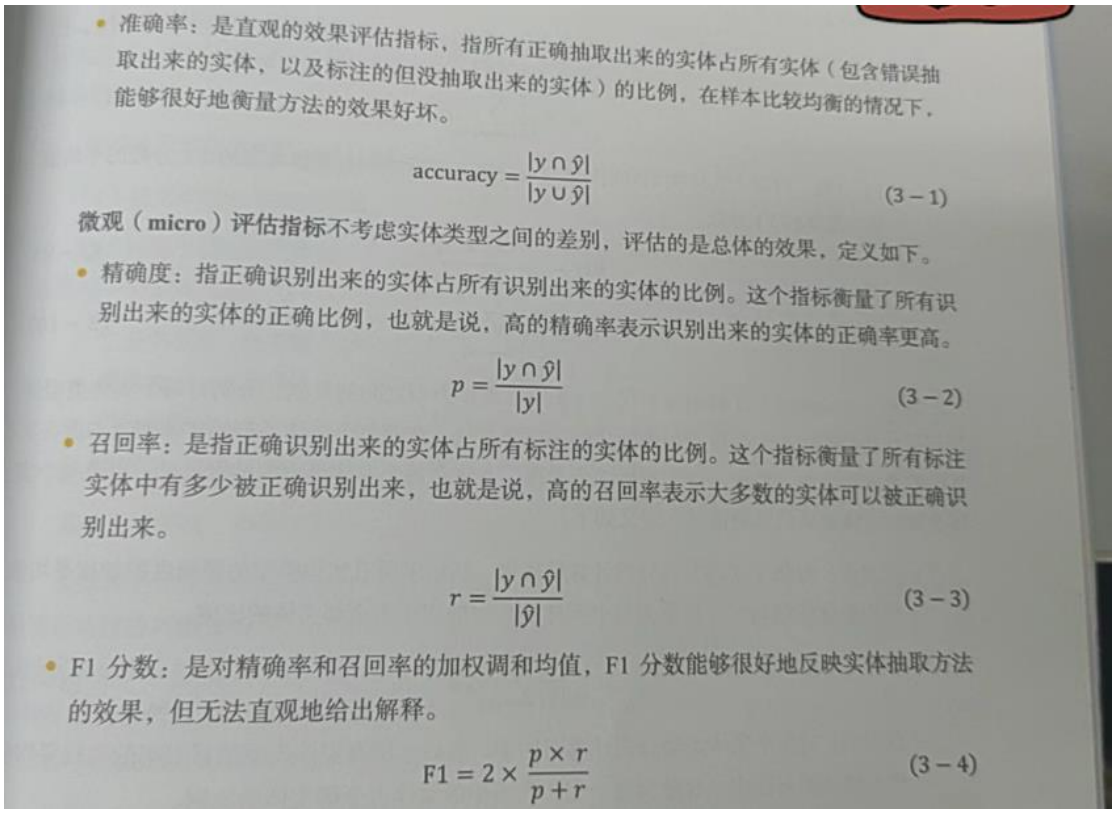
评价指标：准确率、召回率和 F1 分数。

y：所有抽取出来的实体集合，包含实体标签，即抽取的（实体，实体类型）对的集合。

$\check{y}$ ：所有标注的实体集合，包含实体标签，即标注的（实体，实体类型）对集合。

$y \cap \check{y}$ ：表示识别正确的实体，即标注的实体和抽取的实体相同，实体类型也相同。

$y \cup \check{y}$ ：所有标注实体（实体、实体类型）对和抽取（实体、实体类型）对。



- 准确率：是直观的效果评估指标，指所有正确抽取出来的实体占所有实体（包含错误抽取出来的实体，以及标注的但没抽取出来的实体）的比例，在样本比较均衡的情况下，能够很好地衡量方法的效果好坏。

$$\text{accuracy} = \frac{|y \cap \hat{y}|}{|y \cup \hat{y}|} \quad (3-1)$$

微观（micro）评估指标不考虑实体类型之间的差别，评估的是总体的效果，定义如下。

- 精确度：指正确识别出来的实体占所有识别出来的实体的比例。这个指标衡量了所有识别出来的实体的正确比例，也就是说，高的精确率表示识别出来的实体的正确率更高。

$$p = \frac{|y \cap \hat{y}|}{|y|} \quad (3-2)$$

- 召回率：是指正确识别出来的实体占所有标注的实体的比例。这个指标衡量了所有标注实体中有多少被正确识别出来，也就是说，高的召回率表示大多数的实体可以被正确识别出来。

$$r = \frac{|y \cap \hat{y}|}{|\hat{y}|} \quad (3-3)$$

- F1 分数：是对精确率和召回率的加权调和均值，F1 分数能够很好地反映实体抽取方法的效果，但无法直观地给出解释。

$$\text{F1} = 2 \times \frac{p \times r}{p + r} \quad (3-4)$$

传统机器学习方法

传统的有监督机器学习算法被用在实体抽取上，基本原理是：

（1） 将文本划分为 token 序列

（2） 对每个 token 进行分类，分类到某一个具体的实体类型上，或者不属于任何一个实体类型。

机器学习中的分类算法：决策树（最早被用于实体抽取的机器学习算法之一），支持向量机（SVM）,使用了核函数的支持向量机。

在实体抽取领域，概率图模型是使用最多的方法，概率图模型可以对带有依存关系的序列进行建模，是将概率论和图论相结合的机器学习方法。概率图模型的典型算法有：朴素贝叶斯、最大熵模型、隐马尔可夫模型（HMM）、条件随机场（CRF）。

(1) 概率图模型

(2) 朴素贝叶斯模型

(3) 最大熵模型

(4) 隐马尔可夫模型

(5) 条件随机场

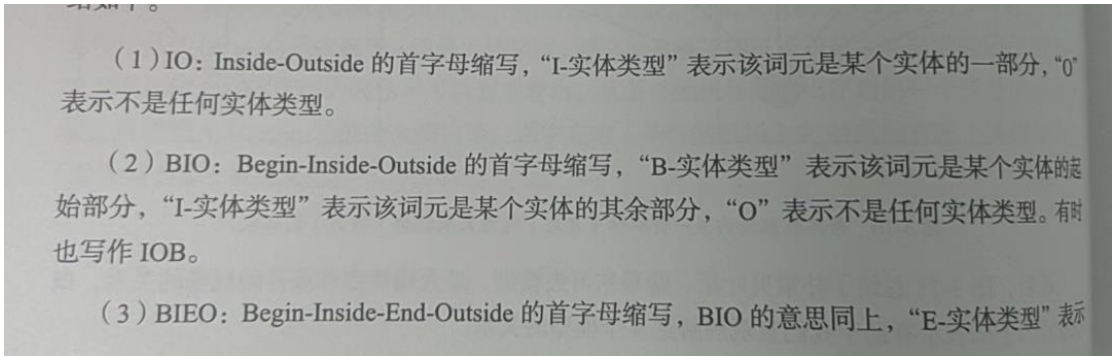
条件随机场是最常用的实体抽取方法之一，是一种无向概率图模型，能够高效处理完全的、非贪婪的、有限状态的推断和训练，特别适合 NLP 领域的分词、词性标注和实体抽取等任务。条件随机场现在也是一个好的基准模型。

在许多序列建模任务（如实体抽取、语义角色标注）的深度学习模型中，条件随机场是常用的解码方法，它可以从全局视角计算 token 之间的依赖关系，从而获得最佳效果。

（6）标记方法

使用机器学习或深度学习方法抽取实体，本质上是一个序列标注问题。序列标注是指使用算法为每个元素打上标记，实现模式识别任务。对于实体抽取任务，就是给每个 token 打上能够标识实体类型的标记。

常见的标记方法：IO、BIO、BIEO、BIESO。



（1）IO：Inside-Outside 的首字母缩写，“I-实体类型”表示该词元是某个实体的一部分，“O”表示不是任何实体类型。

（2）BIO：Begin-Inside-Outside 的首字母缩写，“B-实体类型”表示该词元是某个实体的起始部分，“I-实体类型”表示该词元是某个实体的其余部分，“O”表示不是任何实体类型。有时也写作 IOB。

（3）BIEO：Begin-Inside-End-Outside 的首字母缩写，BIO 的意思同上，“E-实体类型”表示某个实体的结束词元，“I-实体类型”则表示除起始词元和结束词元之外的其他部分。有时也写作 IOBE。

（4）BIESO：Begin-Inside-End-Single-Outside 的首字母缩写，BIEO 的意思同上，“S-实体类型”表示某个单独词元就是一个实体，有时也写作 IOBES。另外，BIESO 也有如下别名。

- BIEOU（Begin-Inside-End-Outside-Unigram），其中 U 等价于 S。
- BILOU（Begin-Inside-Last-Outside-Unigram），其中 L 等价于 E。
- BMEWO（Begin-Middle-End-Word-Outside），其中 M 等价于 I，W 等价于 S。

表 3-2　使用 4 种不同标记方法对同一个样本进行标记的示例

文本	IO	BIO	BIOE	BIESO
11	I-DT	B-DT	B-DT	B-DT
月	I-DT	I-DT	I-DT	I-DT
03	I-DT	I-DT	I-DT	I-DT
日	I-DT	I-DT	E-DT	E-DT
长	I-OBJ	B-OBJ	B-OBJ	B-OBJ
征	I-OBJ	I-OBJ	I-OBJ	I-OBJ
五	I-OBJ	I-OBJ	I-OBJ	I-OBJ
号	I-OBJ	I-OBJ	E-OBJ	E-OBJ
在	O	O	O	O
文	I-LOC	B-LOC	B-LOC	B-LOC
昌	I-LOC	I-LOC	E-LOC	E-LOC
发	I-LOC	B-LOC	B-LOC	B-LOC
射	I-LOC	I-LOC	I-LOC	I-LOC
场	I-LOC	I-LOC	E-LOC	E-LOC
点	O	O	O	O
火	I-OBJ	B-OBJ	B-OBJ	S-OBJ
升	O	O	O	O
空	O	O	O	O

（7）用 CRF++进行实体抽取

从头开始实现一个条件随机场的工作量很大，但有现成工具可用，其中 CRF++是最知名且经典的。

scikit-learn 是一个机器学习工具包，提供了多种机器学习模型，也提供了评估模型结果的方法库。

深度学习方法

深度学习指应用多层神经网络从大量的数据中学习出规律和知识，并进行预测和决策。

在具备较多标注数据的情况下，基于深度学习的实体抽取方法已经成为主流。

基于深度学习的通用实体抽取框架

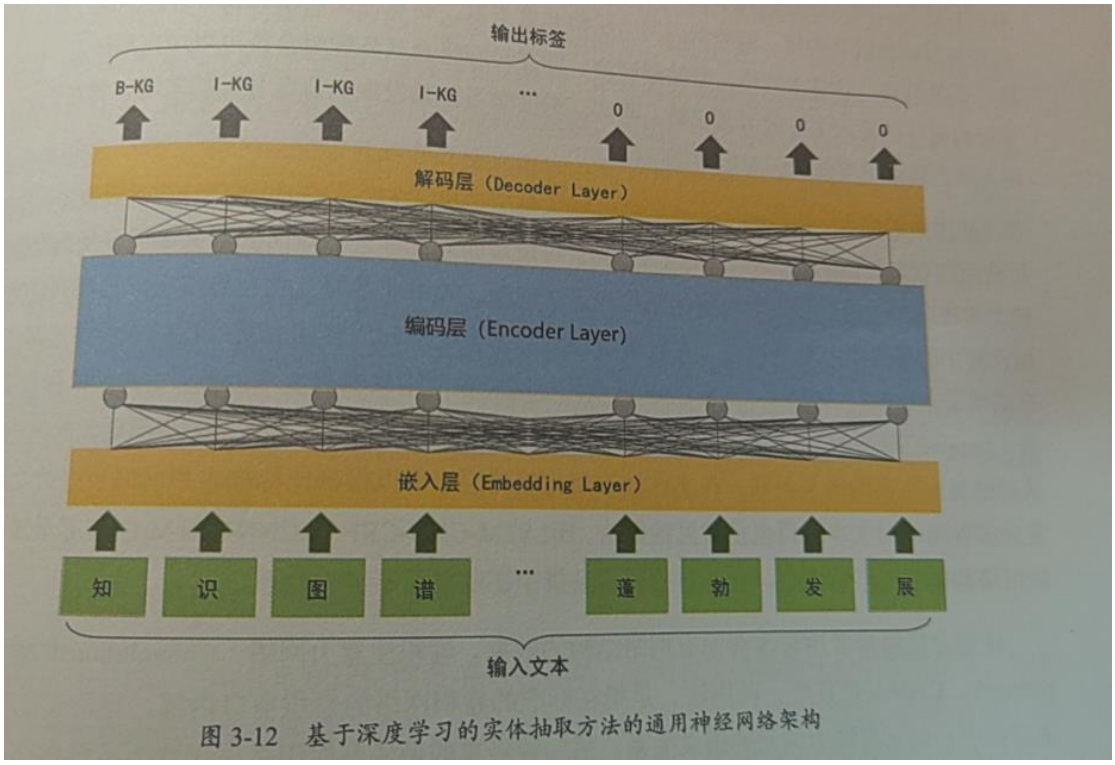


图 3-12　基于深度学习的实体抽取方法的通用神经网络架构

抽象为三层的通用神经网络架构，如上图所示。

从输入文本开始，

第一层：是嵌入层，将组成文本的词元 token 转换为向量表示，为编码层提供向量形式的输入；

第二层：编码层，通过各种复杂网络结构学习输入文本所蕴含的结构和含义，并将学习出的向量表示输出给解码层；

第三层：解码层，按照目标任务对向量进行解码，输出每个词元的标签。

输入文本被分词器切分成 token 序列，并通过映射转换为 token 对应的 id，对于机器学习，token 对应的 id 通常使用独热编码，每个向量的维度都是总词元的个数，对于中文来说，独热编码的维度往往高达数万维或数十万维，英文等语言甚至高达数百万维；对于神经网络，这么高纬度不适合神经网络的计算，需要通过嵌入（embedding），把 token 对应的 id 转换为向量，把维度降到数十维或数百维。token 的嵌入有两种做法，词嵌入和字嵌入，区别在于分词器在分词时是按词切分还是按字切分。

（1）嵌入层：

嵌入层可以使用其它程序或方法预先训练好的数据，并在模型中直接使用查表的方法来实现嵌入运算，这被称为预训练的词向量或字向量。预训练通常会利用大规模语料的无监督学习来训练模型，从而得到更加全面的语义，大多数情况下效果更好。典型模型：Word2vec，ELMo，BERT，ERNIE，GPT3，盘古α。这些预训练模型可以作为查找表，用在实体抽取网络架构的嵌入层中。有些模型会把多个 token 的嵌入相加或者 concat 使用，比如针对一批语料，用 word2vec 分别训练处字向量和词向量，在模型中将字向量和词向量拼接使用，一些模型有特殊的嵌入，比如 Transformer 用到位置编码（positional embedding）。

嵌入层也可以在模型训练过程中学习出来，被称为“跟随训练”。

（2）编码层：

编码层通过不同的网络结构进一步学习输入文本，理解输入文本所蕴含的语义，并针对目标任务进行适配。嵌入层的每个 token 对应一个向量，而编码层将整串输入文本编码为一个适应当前任务目标的语义向量。（语义向量，包含了文本在语义上的信息）

CNN、RNN、前馈网络、残差网络、注意力网络、Transformer、GCN 及其各种组合都可以用在编码层，以更好学习输入文本的语义向量。

BiLSTM-CRF 模型：序列建模领域的基准模型，编码层使用双向长短期记忆网络 BiLSTM，对输入文本进行双向的语义编码，从而能够理解文本从前向后和从后向前的结构与语义依赖。

CNN-CRF 模型：用标准卷积网络来编码文本。除了标准卷积网络，迭代扩张卷积网络（ID-CNN）可以在编码中获得更大的感受野，来捕捉远距离的语义关联，在编码层可以得到比标准卷积网络更好的效果。

BiLSTM-CNN-CRF、CNN-LSTM-CRF：复合多种网络对文本编码也常见。

除了上面的模型，注意力机制可以用来改善原有网络结构的效果，比如，卷积注意力网络，在标准卷积网络的卷积窗口内部，使用注意力机制来捕捉中心 token 和周边 token 的语义关系；另一种做法是使用全局注意力捕捉 BiLSTM 或 BiGRU

网络中的全局语义信息。巅峰就是 Transformer，完全使用自注意力机制来捕捉文本的语义关系，效果远超前面几种模型。基于 Transformer 的变种实现的预训练模型：BERT、ERNIE、GPT-3 等。

图神经网络结构被用于实体抽取，代表性：图卷积网络（Graph Convolutional Network）和图注意力网络（Graph Attention Network）等。

（3）解码层：

编码层的核心是深入理解文本的语义和结构，学习出表示该文本的向量，并输出给解码层，解码层根据目标任务对向量进行解码。

实体抽取任务中，解码层的任务是解码出每个 token 的标签。解码层常用的：条件随机场算法，softmax。

越来越多的模型使用 softmax 代替条件随机场实现实体抽取的解码，优点是需要算力少、计算效率高、响应迅速。

不常用：维特比解码，指针网络。

BiLSTM-CRF 模型

BiLSTM-CRF（双向长短期记忆网络-条件随机场）模型在实体抽取任务中用的最多，是实体抽取任务深度学习模型的评测基准，是 BERT 出现之前最好用。

BiLSTM-CRF 不需要利用特征工程，通过 BiLSTM 网络自动从数据（训练语料）中学习出特征，并通过 CRF 计算标签的全局概率信息对输出 token 序列向量进行解码，得到对应的标签序列。

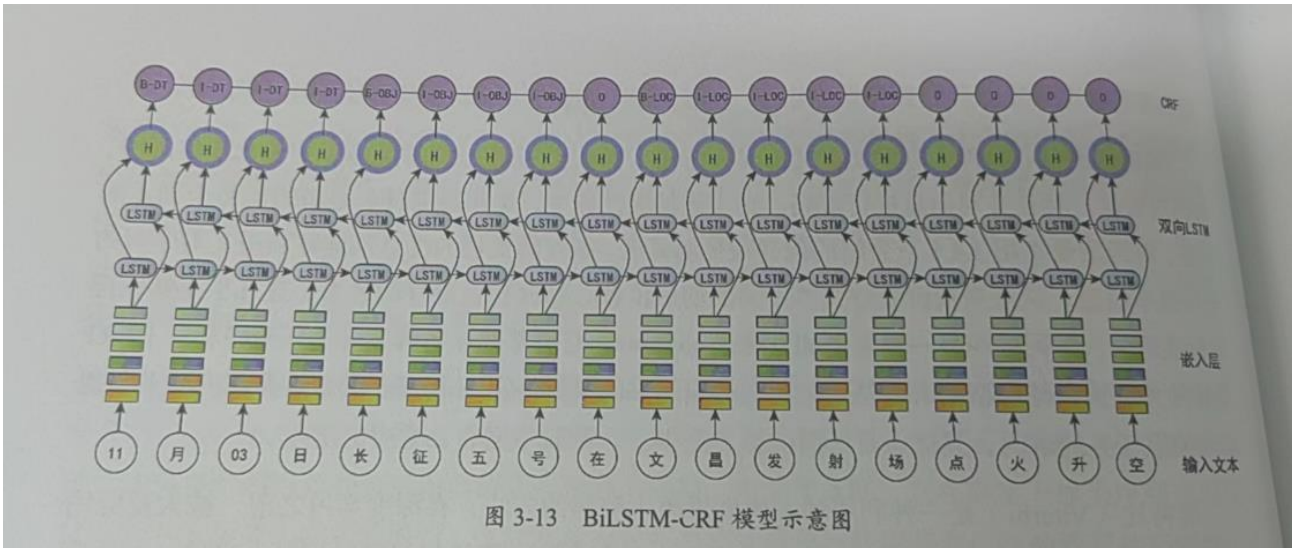


图 3-13　BiLSTM-CRF 模型示意图

总体：在嵌入层，把输入文本序列通过分词器转化为 token 序列，嵌入层把 token 序列转化为向量。（把文本转化为向量由两种方法：在训练模型的时候同时训练出 token 的向量表示；使用预训练的 token 向量，方法由 word2vec、Glov，优点是可以从大规模文本中学习出更好的语义表示，泛化能力更强）在编码层，使用前向 LSTM 和后向 LSTM 分别对输入文本从前往后和从后往前进行编码，学习出文本的语义向量和结构，最后由 CRF 计算全局概率信息来解码输出的标签序列。

预训练模型

预训练模型是指用无监督学习或多任务学习的方法从大规模语料中训练出来的通用模型，逻辑是：如果一个模型是基于足够大且通用（领域内通用）的数据集训练出来的，那么这个模型能够充分学习出这个数据集分布的特征和规律。

预训练模型已经具备了通用的知识（或领域内通用的知识），从而更好实现模型目标。

使用预训练模型的方法：

（1） 直接使用原有模型的网络结构，并载入预训练模型，然后使用新的数据对模型进行 fine-tuning（微调）。

（2） 为特定任务构建的深度神经网络结构中包含预训练模型的网络结构，在训练时载入预训练模型，并冻结对应网络结构的参数训练。

文本领域的预训练模型：BERT、GPT-3、盘古 α 、ERNIE、ALBERT、RoBERTa、DistilBERT、BigBird、XLNet、XLM、MobileBERT 等。

预训练模型用于实体抽取

预训练模型用于实体抽取的两种方法：

（1） 预训练模型用在嵌入层，把预训练模型当成通用的 token 嵌入来使用。如 BERT-BiLSTM-CRF.

（2） 预训练模型直接取代嵌入层和编码层，即把预训练模型作为实体抽取神经网络模型的主体网络结构，仅在解码层对实体抽取任务进行适配。也是预训练模型结合微调的典型应用。如 BERT-softmax 和 BERT-CRF。

BERT 模型：

BERT 是模拟人类对语言认知的双向语言模型，属于掩码语言模型（MLM）。BERT 在训练模型时，用掩码标识符取代训练语料的文本（即 token 序列）中一定比例（原论文是 15%）的 token，一般来说，被抽出来的掩码部分，80%替换为掩码标记、10%替换为随机 token，另外 10%保持原样不变。

BERT 是用的是 Transformer 的编码器结构，核心是多头注意力机制，除了多头注意力机制，为了实现语言序列中不同 token 位置具有不同的语义，BERT 引入了位置嵌入（positional embedding）；为了实现上下句表达，BERT 引入区分上下句的片段嵌入（segmentation embedding）。

图 3-18 BERT 的嵌入层，由位置嵌入、片段嵌入和词元嵌入相加得到

所以 BERT 的嵌入层由位置嵌入、片段嵌入和 token 嵌入相加得到，

利用 BERT 预训练模型，并在具体任务上进行微调，在很多自然语言处理任务中达到了最高水平。

弱监督学习方法

为了解决缺少样本的问题，弱监督学习被提出。

https://zhuanlan.zhihu.com/p/541288334

应对人工标注语料成本高最直接的方法是使用算法自动标注语料，引导法是典型的方法之一。还有部分标注。

弱监督学习还包括迁移学习和远程监督等。预训练模型是迁移学习的典型方法，建立在大规模预训练模型之上的少样本学习，单样本学习和零样本学习。迁移学习的另一种典型应用是生成对抗网络（GAN）。

关系抽取

知识图谱中的关系用三元组<头实体，关系，尾实体>来表示，也叫做关系三元组。也被表示为<主语 subject，谓语 predicate，宾语 object>简称 SPO 三元组。

关系抽取是指从非结构化文本中抽取出符合事实的关系三元组，即判定两个实体间是否存在某种语义化的有向关系。

关系抽取分为两种类型：

（1） 开放式关系抽取：实体之间可能存在的关系集合没有预先定义

（2） **封闭式关系抽取**：有一个预先定义的实体之间可能存在的关系集合，比如知识图谱模式中的关系类型列表，目的是判断给定的两个实体是否是关系集合中的一种或者都不是。

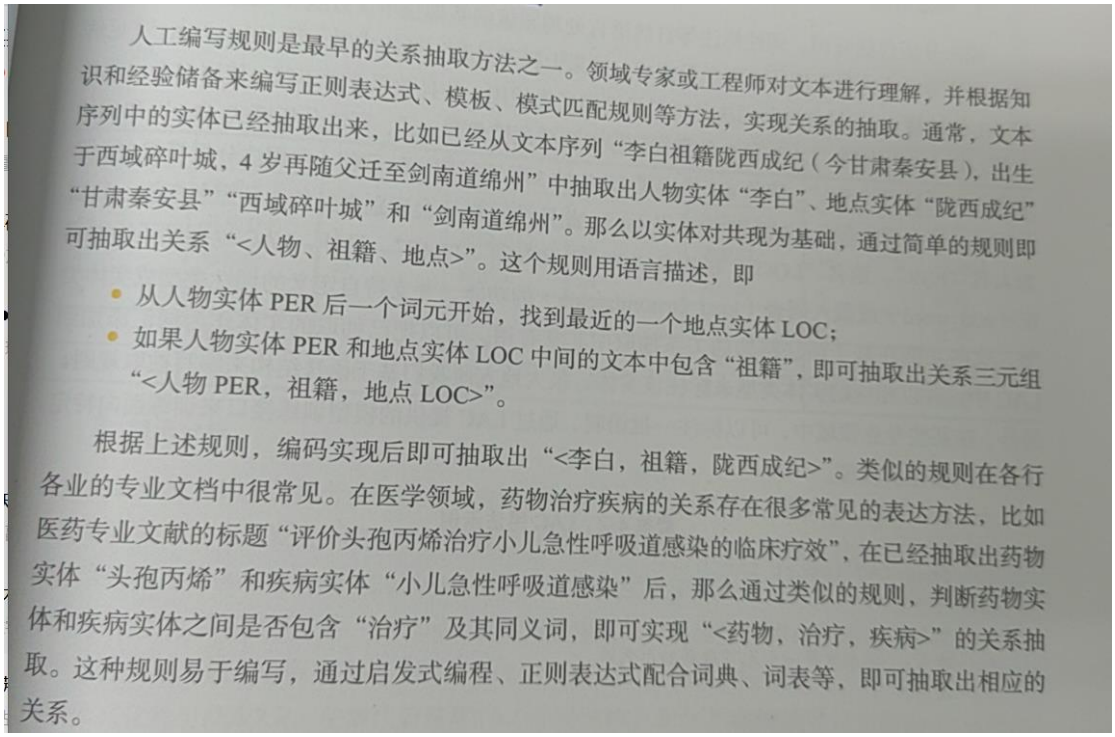
Pipeline 方法：通过实体抽取，每个实体的类型和在文本中的位置已明确，这时关系抽取的目标是判断在一段文本中两个实体的关系是否属于给定的关系类型集合中的一种。就是先抽取实体，再抽取关系。Pipeline 方法的前提是头实体和尾实体同时在文本中出现。

实体-关系联合抽取：给定一个文本序列，在关系抽取任务中需要把实体也抽取出来。

基于规则的关系抽取方法

人工编写规则是最早的关系抽取方法之一。

领域专家对文本进行理解，并根据知识和经验储备来编写正则表达式、模板、模式匹配规则等方法，实现关系抽取。



人工编写规则是最早的关系抽取方法之一。领域专家或工程师对文本进行理解，并根据知识和经验储备来编写正则表达式、模板、模式匹配规则等方法，实现关系的抽取。通常，文本序列中的实体已经抽取出来，比如已经从文本序列“李白祖籍陇西成纪（今甘肃秦安县），出生于西域碎叶城，4 岁再随父迁至剑南道绵州”中抽取出人物实体“李白”、地点实体“陇西成纪”“甘肃秦安县”“西域碎叶城”和“剑南道绵州”。那么以实体对共现为基础，通过简单的规则即可抽取出关系“<人物、祖籍、地点>”。这个规则用语言描述，即

- 从人物实体 PER 后一个词元开始，找到最近的一个地点实体 LOC；
- 如果人物实体 PER 和地点实体 LOC 中间的文本中包含“祖籍”，即可抽取出关系三元组“<人物 PER，祖籍，地点 LOC>”。

根据上述规则，编码实现后即可抽取出“<李白，祖籍，陇西成纪>”。类似的规则在各行各业的专业文档中很常见。在医学领域，药物治疗疾病的关系存在很多常见的表达方法，比如医药专业文献的标题“评价头孢丙烯治疗小儿急性呼吸道感染的临床疗效”，在已经抽取出药物实体“头孢丙烯”和疾病实体“小儿急性呼吸道感染”后，那么通过类似的规则，判断药物实体和疾病实体之间是否包含“治疗”及其同义词，即可实现“<药物，治疗，疾病>”的关系抽取。这种规则易于编写，通过启发式编程、正则表达式配合词典、词表等，即可抽取出相应的关系。

基于规则的另一个方法是编写模板，特别是有固定结构的网页上。模板的编写方法和实体抽取类似。

词法分析和依存句法分析

关系三元组本身就是从句子的主语、谓语和宾语直接映射过来，如果有工具可以很好解析句子的语法结构，那么基于语法结构编写规则来抽取关系就水到渠成了。且，NLP 领域经过多年的发展，现在的词法分析和句法分析工具已经很成熟了，并成功用在关系抽取上。

词法分析

词法分析包括分词、词性标注等技术。

工具：Jieba、LAC（https://gitee.com/baidu/lac）

在某些专业领域，可以标注一批语料，通过 LAC 提供的模型训练接口来训练面向特定场景的词法分析模型。

句法分析

句法分析，又叫做句子成分分析或句法结构分析。

依存句法分析是现在使用最多的一种概率语法分析方法，把句子解析为由词语组成的依存树，并标记了词与词之间的关系。

基于深度学习的依存句法分析效果已是当前主流的方法。DDParser 是一个效果不错的基于深度学习的中文句法依存分析工具，调用 LAC 对句子进行分析和习性标注。

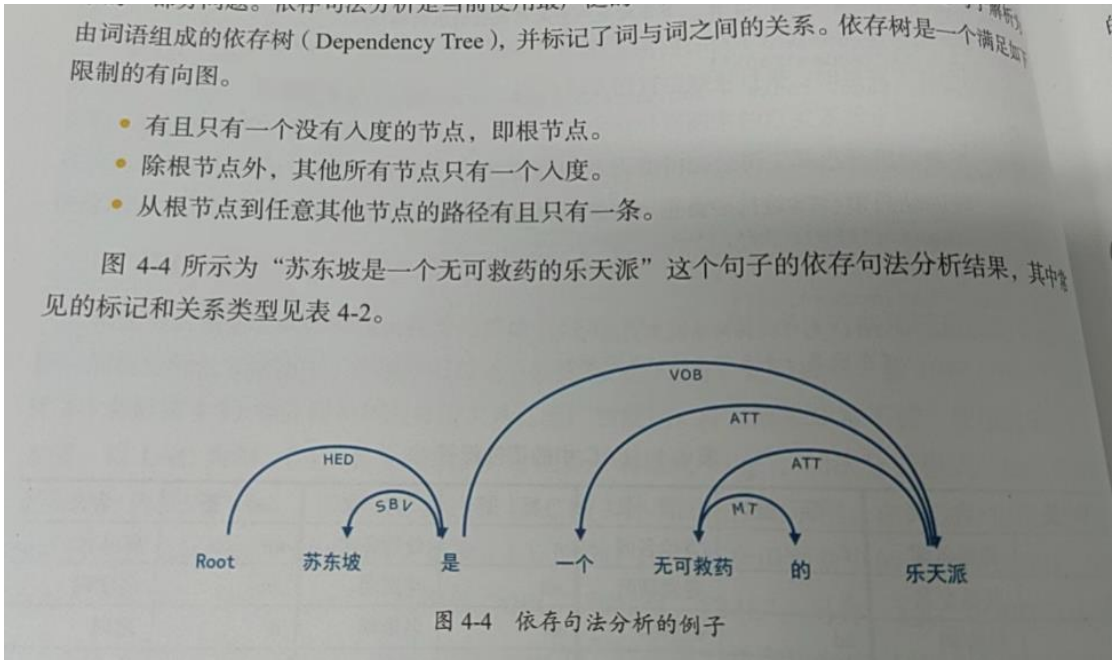


表 4-2　句法分析中常见的标记和关系类型

标　记	关系类型	标　记	关系类型
SBV	主谓关系	COO	并列关系
VOB	动宾关系	DBL	兼语结构
POB	介宾关系	DOB	双宾语结构
ADV	状中关系	VV	连谓结构
CMP	动补关系	IC	子句结构
ATT	定中关系	MT	虚词成分
F	方位关系	HED	核心

一个句子通常由主干和枝叶组成。这个句子的主干成分包括主语“苏东坡”、谓语“是”和宾语“乐天派”，图 4-4 的依存树由主谓关系 SBV——“苏东坡/是”和动宾关系 VOB——“是/乐天派”组成。句子中除主干之外的剩余部分是句子的枝叶，在图 4-4 中的例子中包括了两个定语“一个”和“无可救药的”，它们分别和宾语的“乐天派”组成了定中关系 ATT。进一步的，定语“无可救药的”中的“的”字是虚词，被分割开后，组成了虚词成分关系 MT。

基于语法结构的关系抽取

关系三元组和语法结构关系密切，也有成熟的词法和句法分析工具，所以基于语法结构的关系抽取逐渐流行。

抽取过程：首先，使用词法分析工具对文本进行分词和词性标注，并把已抽取的新词作为实体添加到词法分析工具中，来保证能完整地把实体切分为一个整体，并且被标注上合适的“词性”。然后，使用句法分析工具解析文本序列的依存句法树，为词与词之间标注合适的语法关系，提取除句子的主语、谓语、宾语、定语等语法成分的内容。最后，结合知识图谱模式对实体间关系的约束，利用语法结构编写合适的关系抽取规则，完成对关系三元组的抽取。

句子“中国科学院院士吴文俊荣获2000年度首届国家最高科学技术奖”是图4-3所示文本的一个简化版，图4-5是使用LAC和DDParser解析的依存句法树，根据句法结构中的主谓宾可以抽取出三元组<吴文俊，获得，国家最高科学技术奖>。

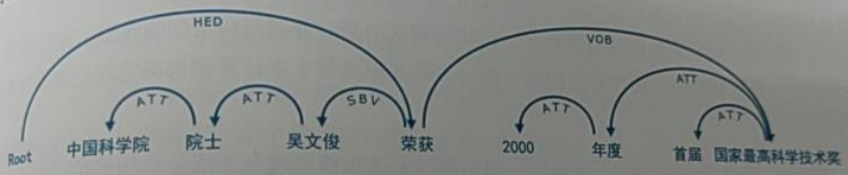


图4-5 使用LAC和DDParser解析的依存句法树（一）

以抽取关系类型为“<人物，获得，奖项>”的三元组为例。

（1）根据解析出来的依存句法树，通过主谓关系SBV和动宾关系VBO的句子成分，能够提取出句子的主语、谓语和宾语，分别得到“吴文俊”“荣获”和“国家最高科学技术奖”。

（2）利用同义词典判断“荣获”和“获得”是同义词。

（3）结合“吴文俊”和“国家最高科学技术奖”的实体类型分别是人物和奖项，判断这个句子的主谓宾所组成的SPO三元组恰好符合关系类型“<人物，获得，奖项>”。

（4）断定三元组“<吴文俊，获得，国家最高科学技术奖>”是所要抽取的关系三元组。

基于语法结构的关系抽取步骤：

基于这个例子，可以总结出基于语法结构的关系抽取的步骤，具体如下。

（1）将已抽取实体作为新词，添加到词法分析工具中，词性可设定为实体类型。

（2）对文本序列进行词法分析，确保实体被完整地切分为完整的一个词元，并且正确地将词性标注为实体类型。

（3）结合知识图谱模式或关系类型的约束，确定实体对可能的关系，以及实体对在句子中的句法关系。

（4）基于语法编写规则，抽取能够表达关系的关键词。

（5）通过同义词典、业务词表、语义相似性工具等，判断表达关系的关键词是否与关系类型中描述关系的关键词相匹配。

（6）判断关系三元组是否成立。

成熟的语义相似度计算工具：基于词向量（word2vec、GloVe）、基于大规模预训练模型BERT、GPT、ERNIE的语义相似度计算工具。这些工具使用词向量或预训练模型将文字转化为稠密向量，并通过计算向量间的距离来计算词与词之间是否表达相同的意思。

基于深度学习的关系抽取方法

专门针对关系抽取的深度学习模型能从数据中学习出更丰富和完善的规则，效果更好。

关系分类

封闭式的关系抽取中，关系受知识图谱模式中关系类型的限定；Pipeline 方法中实体已经被抽取出来。在这两个条件限定下，关系抽取也被叫做关系分类。例子，关系类型有<人物，颁发，奖项>〈人物，获得，奖项>〈人物，命名，奖项>，输入文本是“中国科学院院士吴文俊荣获 2000 年首届国家最高科学技术奖”和已抽取的两个实体<吴文俊，人物>〈国家最高科学技术奖，奖项>，关系分类的目标就是判断其是否属于上述 3 个关系类型中的某一个，或者不是其中任何一种关系。

关系分类中也使用了大规模的预训练模型

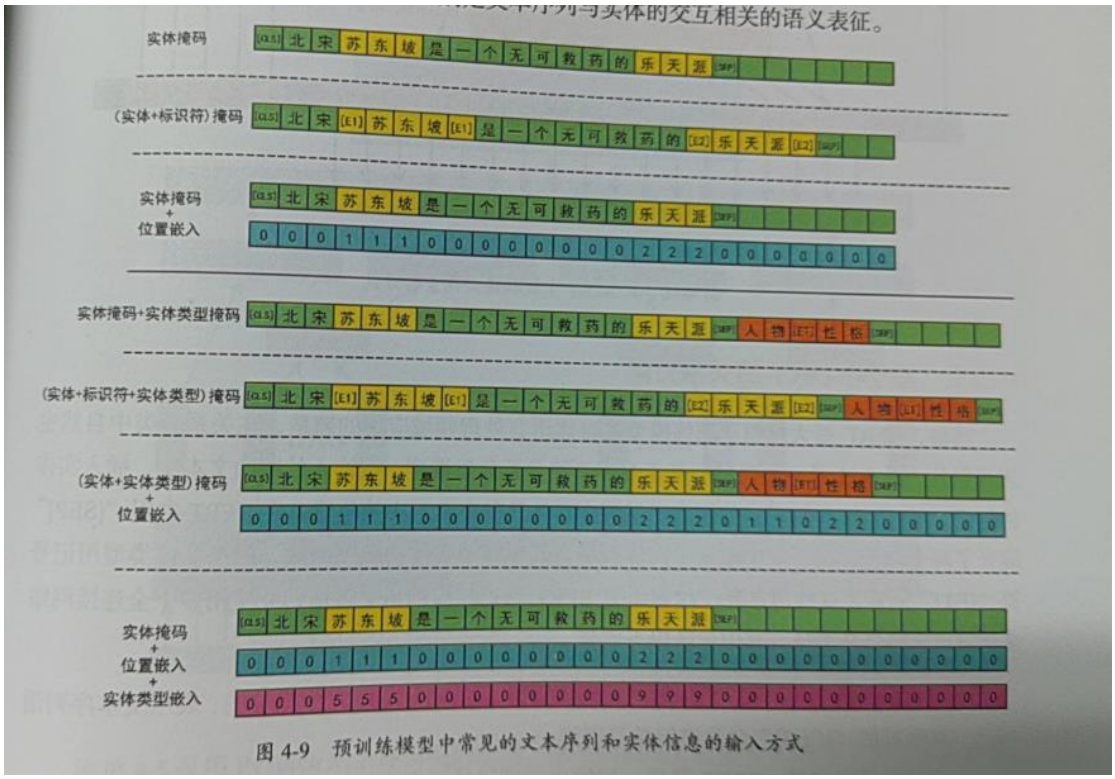


图 4-9 预训练模型中常见的文本序列和实体信息的输入方式

基于 BERT 的关系分类

典型的基于 BERT 的关系分类模型:

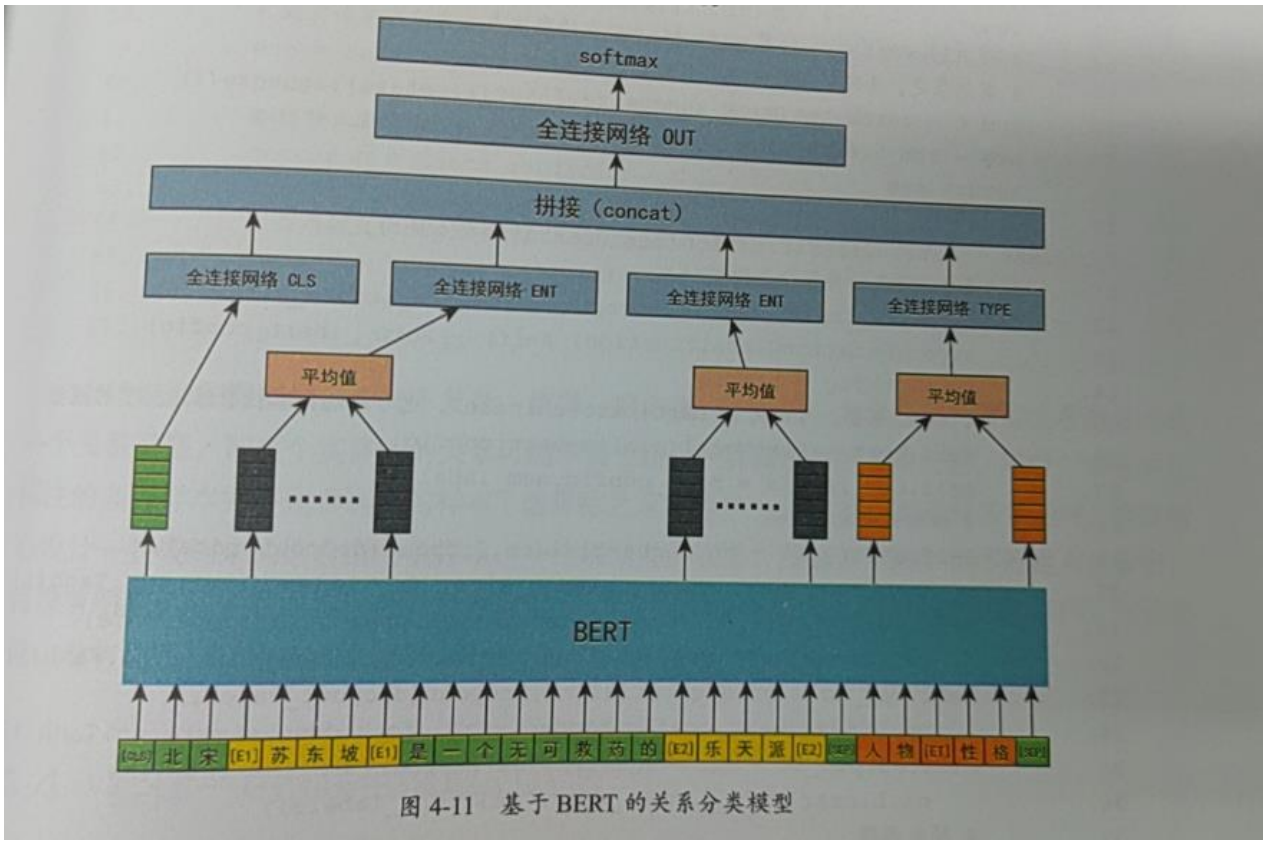
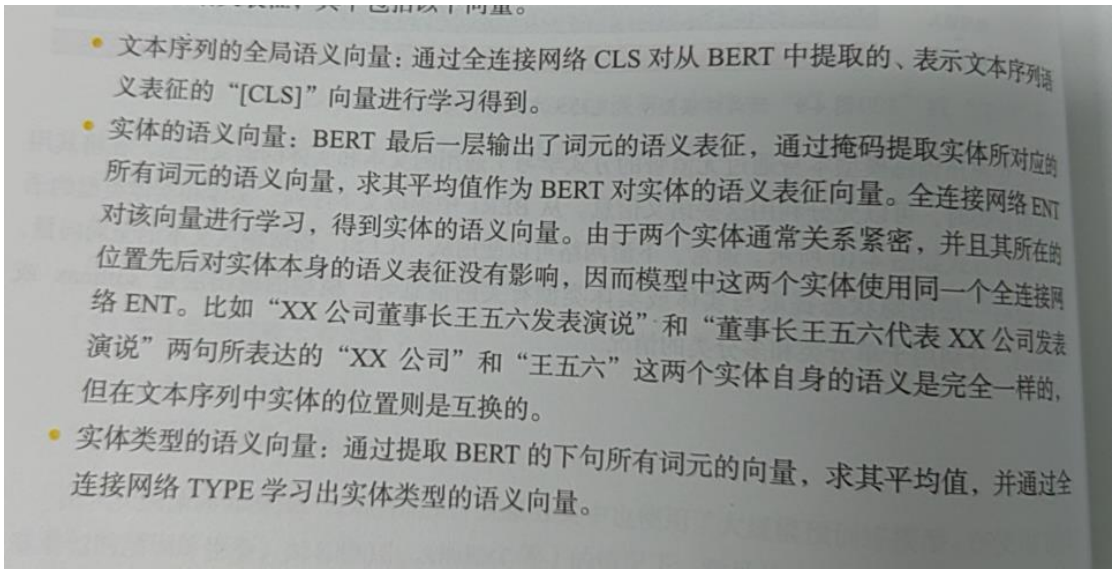
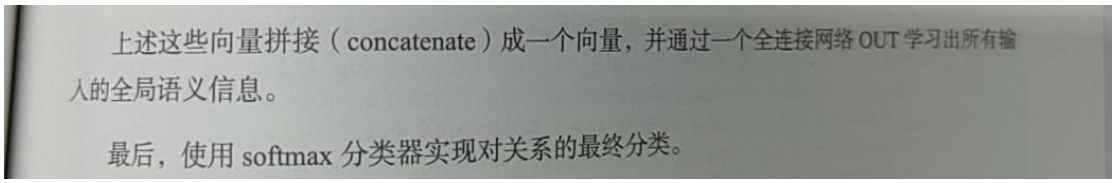


图 4-11　基于 BERT 的关系分类模型

实体标记[E1]和[E2]，[SEP]隔开两个句子，实体类型用[ET]隔开。



- 文本序列的全局语义向量：通过全连接网络 CLS 对从 BERT 中提取的、表示文本序列语义表征的“[CLS]”向量进行学习得到。
- 实体的语义向量：BERT 最后一层输出了词元的语义表征，通过掩码提取实体所对应的所有词元的语义向量，求其平均值作为 BERT 对实体的语义表征向量。全连接网络 ENT 对该向量进行学习，得到实体的语义向量。由于两个实体通常关系紧密，并且其所在的位置先后对实体本身的语义表征没有影响，因而模型中这两个实体使用同一个全连接网络 ENT。比如“XX 公司董事长王五六发表演说”和“董事长王五六代表 XX 公司发表演说”两句所表达的“XX 公司”和“王五六”这两个实体自身的语义是完全一样的，但在文本序列中实体的位置则是互换的。
- 实体类型的语义向量：通过提取 BERT 的下句所有词元的向量，求其平均值，并通过全连接网络 TYPE 学习出实体类型的语义向量。


上述这些向量拼接（concatenate）成一个向量，并通过一个全连接网络 OUT 学习出所有输入的全局语义信息。

最后，使用 softmax 分类器实现对关系的最终分类。

实体-关系联合抽取的方法

实体-关系联合抽取方法
基于片段预测的实体-关系联合抽取

弱监督学习和关系抽取

虽然基于深度学习的方法在关系抽取上表现出了非常好的效果，但因需要大量的标注数据，导致在实际应用中面临诸多挑战。很多场景下没有大量的样本，比如在一些专业领域可能只有少量的数百份或数千份文档，但多数深度学习模型需要数万份或更多标注数据来能达到较好的效果。
为了应对这些挑战，就有了弱监督学习。目的是充分挖掘少量已标注样本的潜力，实现少样本下更好的效果。

引导法

远程监督

弱监督学习与 Snorkel

知识存储

知名度高、使用面广的分布式图数据库 JanusGraph，

本章所讲解的图数据库的对比如表 5-6 所示。

表 5-6　图数据库对比一览表

	JanusGraph	Neo4j	Dgraph	NebulaGraph
首次发布	2017 年	2007 年	2016 年	2019 年
开发语言	Java	Java	Go	C++
属性图模型	完整的属性图模型	完整的属性图模型	不完整的属性图模型，更接近于 RDF 存储	完整的属性图模型
架构	分布式	单机	分布式	分布式
存储后端	Hbase 、Cassandra 、BerkeleyDB	自定义文件格式	键值数据库 BadgerDB	键值数据库 RocksDB

续表

	JanusGraph	Neo4j	Dgraph	NebulaGraph
物理存储	KCV	KV	KV	KV
高可用性	支持	不支持	支持	支持
高可靠性	支持	不支持	支持	支持
一致性协议	HBase：Paxos；Cassandra：基于多数派（Quorum-based）	无	RAFT	RAFT
跨数据中心复制	支持	不支持	支持	不支持
事务	BerkeleyDB：完全的 ACID 支持；HBase 和 Cassandra：BASE，通过锁和两阶段提交能够实现更强的一致性保证	完全的 ACID	基于 Omid 修改版的分布式事务	不支持分布式事务
分区策略	随机分区，支持显式指定分区策略	不支持分区	自动分区，自动再平衡，再平衡时会拒绝写入和更新	哈希（取模）静态分区，分区数设定后不能更改

分区方法	根据顶点 id 分区，每边存储两次	不支持分区	更新	分区数设定后不能更改
大数据平台集成	Spark、Hadoop、Giraph	Spark	根据边标签（谓词）分区	根据顶点 id 分区，每条边存储两次
顶点标签	0 个或 1 个	0 个或多个	不支持	Spark、Flink
顶点间相同标签的多条边	支持，并且支持多种约束条件，包括 ONE2ONE、ONE2MANY、MANY2ONE、MULTI、SIMPLE	顶点对之间支持多条相同标签的边	0 个	1 个或多个
查询语言	Gremlin，通过 cypher-for-gremlin 可支持 openCypher	Cypher，通过插件可支持 Gremlin、GraphQL 等	不支持	顶点对之间支持多条相同标签的边
全文检索	ElasticSearch、Solr、Lucene	内置	GraphQL	nGQL
多个图	支持创建任意多图	一个实例只能有一个图	内置	ElasticSearch
属性图模式	无模式，可选模式约束，强制模式约束	可选模式约束	一个集群只能有一个图	支持创建任意多图
客户端协议	HTTP、WebSockets	HTTP、BOLT	无模式	强制模式约束
客户端语言	Java、Python、C#、Go、Ruby、Rust	Java、.NET、JavaScript、Python、Go	HTTP、gRPC、Protocol Buffer	HTTP
			Java、JavaScript、Go、Python、.Net	Python、Java、Go、C++