# 文本分析与文本挖掘

姜　维/著

科学出版社

# 文本分析与文本挖掘

姜 维 著

国家自然科学基金出版支持

科 学 出 版 社

北 京

## 内 容 简 介

本书阐述词法分析、文本分类、文本聚类、文本检索、垃圾邮件过滤、情感分析、个性化推荐等文本分析与文本挖掘方面的理论方法。人工智能技术与互联网的发展更是为该领域研究提出新的需求，书中相关理论和技术可以直接用于解决具体文本分析与文本挖掘的问题，也可以为进一步研究提供理论方法基础。本书包括理论、技术，既适合理论方法的学习，又适合工程实践。本书配套软件、更多案例、技术文档、配套 PPT 课件等请登录 http: //www.jiangw.cn 和 http: //www.jiangw.com 查询。

本书可作为文本分析与文本挖掘研究人员参考用书，也可作为相关专业的研究生和高年级本科生教学用书。

**图书在版编目（CIP）数据**

文本分析与文本挖掘 / 姜维著. —北京：科学出版社，2018.11

ISBN 978-7-03-059120-3

Ⅰ. ①文… Ⅱ. ①姜… Ⅲ. ①数据采集-研究 Ⅳ. ①TP274

中国版本图书馆 CIP 数据核字（2018）第 237924 号

责任编辑：陈会迎 / 责任校对：孙婷婷

责任印制：吴兆东 / 封面设计：无极书装

科 学 出 版 社 出版

北京东黄城根北街 16 号

邮政编码：100717

http：//www.sciencep.com

北京虎彩文化传播有限公司印刷

科学出版社发行 各地新华书店经销

*

2018 年 11 月第 一 版 开本：720 × 1000 1/16

2018 年 11 月第一次印刷 印张：15 3/4

字数：300 000

**定价：110.00 元**

（如有印装质量问题，我社负责调换）

# 前　言

随着机器学习、数据挖掘、互联网技术等方面的快速发展，文本分析与文本挖掘理论方法的研究和应用已成为前沿热点问题，在多个领域已经取得有价值的研究成果。互联网下的电子商务、商品评论、股市论坛、网络新闻、自媒体、网络舆情、商业情报收集等更是为文本分析与文本挖掘研究提出了新的研究课题。

本书出版得到国家自然科学基金（No.71671052，No.71271066，No.70801022）的支持，作者基于多年在文本分析与文本挖掘方面的成果积累，结合该方向的发展撰写本书，其目的主要包括：①为高校开设的“文本分析与文本挖掘”课程提供一本知识内容较系统的书籍；②为后续《高级文本分析与文本挖掘》一书提供支持；③为相关科研人员提供理论方法参考。

作者基于前述目的精心选择和设定书中内容，本书主要特点包括以下几个方面。

（1）可作为大学高年级本科生和研究生的授课教材。本书兼顾知识深度和系统性，在叙述典型的理论方法基础上，有些章节给出较新的研究成果，有些章节顾及知识的系统性，给出若干前沿研究方向和研究方法说明，有些章节还适当补充了基础模型，如第 9 章中朴素贝叶斯分类的例子。

（2）系统性阐述理论方法的工作原理、具体工作过程和问题案例，着重于文本分析与文本挖掘相关内容的阐述。统计分析、机器学习、数据挖掘中的理论方法在文本分析与文本挖掘中有着重要应用。书中着重阐述和文本分析与文本挖掘密切相关的内容，而对于一些相对通用性的理论方法知识则可参考作者的另一本著作《数据分析与数据挖掘》，其包括通用性数据分析和数据挖掘的理论方法细节。

（3）理论与实践相结合讲述。一方面，重点阐述文本分析与文本挖掘的理论方法，并将该方法与实际问题结合起来；另一方面，书中的理论方法易于实践，可借助软件工具或自己编程实现，此外本书还提供配套软件库，包括若干数据分析与数据挖掘算法和文本分析与文本挖掘算法，支持 C++、C 和 Delphi 等编程语言接口，有助于快速学习和科研。

文本分析与文本挖掘是一个正在快速发展的研究方向，一些前沿问题、新的理论方法、新技术也将层出不穷，令人向往。《高级文本分析与文本挖掘》将在更高一层主题上开展研究和讨论，其内容是本书的延续和扩展。

感谢课题组成员的支持。本书撰写中也参考了国内外同行的研究成果，特别

是一些基础理论方法都是国内外众多科研人员努力的成果，在此表示感谢。本书得到哈尔滨工业大学管理学院的基金资助。文本分析与文本挖掘研究领域的需求不断变化，理论方法也持续发展，书中难免存在不足之处，敬请各位专家与学者批评指正。

配套网站上共享着技术资料、书籍勘误表、最新研究文档、常见问题、在线研讨、作者联系方式等。网址：http: //www.jiangw.cn、http: //www.jiangw.com。

姜　维

哈尔滨工业大学

2018 年 1 月

# 目　录

# 第1章　统计中文分词技术

词法分析是自然语言处理技术的基础，其性能将直接影响句法分析及其后续应用系统的性能。本书的中文词法分析主要包括自动分词、词性标注和中文命名实体识别三个方面，而本章将阐述中文（汉语）自动分词技术。在许多中文文本分析与文本挖掘中，分词往往是第一步工作。中文的词是能够独立运用的最小的语言单位，一般来说，一个词有明确的语义表达，正确的分词是后续语言分析和处理的一项重要前序工作。在不考虑上下文情况下，就单个词来说，可能存在一词多义现象，这就需要后续进一步的语言分析来识别具体的语义。

## 1.1　词法分析问题

### 1.1.1　词法分析研究的问题

分词是指对于中文语句进行各个词的分隔，通常以语句为单位进行各个词的分离。例如，“我要好好学习文本分析与文本挖掘。”经过分词后变为“我/要/好好/学习/文本/分析/与/文本/挖掘/。”；“高校生活丰富多彩”经过分词后变为“高校/生活/丰富多彩”。

现代的分词系统已经具有较高的性能，通常能够满足大多数语言分析、文本分析的需求。但对于某些对分词性能有着更高要求的语言处理，分词性能表现出来的局限性仍较大。例如，“市场/中/国有/企业/才/能/发展”，其中的“中/国有”与“中国/有”、“才能”与“才/能”均为歧义切分，在机器翻译应用中，若切分错误可能会导致整个翻译的失败。

词性标注是为语句中的每个词标注其词性，通常以语句为单位标注各个词的词性。例如，“我/要/好好/学习/文本/分析/与/文本/挖掘/。”经过词性标注后变为“我/r 要/v 好好/d 学习/v 文本/n 分析/vn 与/c 文本/n 挖掘/vn 。/w”。在这里，r、v、d、n、vn、c、w 分别代表代词、动词、副词、名词、动名词、连词、标点。

现代的词性标注性能也非常高，通常能满足常见的文本分析与文本挖掘任务。一般来说，词性标注与分词系统由同一词法分析系统完成，这样能够保证分词过程和词性过程的良好衔接，如系统内所存储的词和相应词性有着良好的对应关系。

命名实体是指人名、地名、机构名等。人名如王岩、孙桂平、王二小；地名

如北京、哈尔滨、北京市东城区王府井（大街）；机构名如清华大学、哈尔滨工业大学、中国国际航空股份有限公司。命名实体可看作一个词，若其搭配无法在词法分析系统构建则全部收集，应用命名实体识别技术帮助识别。其他命名实体还包括商品名、武器名等。

命名实体识别技术一方面要研究对应实体类型的命名特点，另一方面要紧密地结合上下文环境做分析。各类命名实体的识别性能既与实体类型有较大关系，也与给定语句的上下文信息的充分性关系密切，有些命名实体识别技术研究甚至结合文本环境，以此来更准确地判别命名实体。

### 1.1.2　词法分析研究面临困难

相比英文词法分析，中文词法分析有着自己的特点。①从中文语言的特点来看，第一，因为中文各词之间不存在显式的分界符，所以中文需额外的分词过程。第二，中文缺少英文中类似-ed、-ing、人名首字母大写等丰富的词形信息，这将导致标注中文词性时可用信息少。而对于命名实体识别来说，上述差别不仅导致实体识别过程缺少英文中丰富的词形信息，如通常英文人名首字母大写，还导致增加额外识别实体边界的任务。②从外在因素来看，中文自然语言处理研究起步较晚，目前还未达到英文所具有的大规模公开的评测机制与规范的评测语料，由此许多学者的研究工作未能在相同标准下对比，不利于共享彼此的研究成果。③从词法分析本身来看，分词面临着切分歧义问题与未知词识别问题；词性标注主要面临复杂兼类词消歧与未知词标注问题；命名实体识别任务不仅需要划分出实体的边界，还需要识别出实体的类型。三者所面临的问题并非孤立，而是相互关联的，因此如何协调地利用彼此的信息，同时有效地完成词法分析的任务是一个亟待探索的问题。

近些年的研究成果表明，现有监督方法在解决词法分析问题时面临着性能瓶颈，对于模型自身的改进并未取得显著的成效。其主要原因有两点：①数据稀疏问题的影响。因为语言中许多统计现象符合 Zipf 定律（即数据出现长尾现象，即使增大语料库仍然面临着某些特征很少出现的现象，因此数据稀疏问题严重），所以这种数据稀疏问题仅通过增大语料库的方式是难以避免的。②应用场合数据与训练数据难以保持独立同分布的条件。在实际使用中，往往不能完全满足应用场合数据与训练数据独立同分布这一条件。在克服第一个问题的影响时，除了模型本身的改进，如在 N-gram 模型中采用平滑算法等，还可从使用特征角度挖掘更有效的特征，以及引入领域知识词典或推理机制。在克服第二个问题的影响上，通常的方法只能是尽可能收集与应用场合数据同源的训练数据。

### 1.1.3　一体化中文词法分析框架

从计算语言角度来看，分词、词性标注、命名实体识别面临着不同的任务。分词可看作序列切分的过程；词性标注则是序列标注的过程；而命名实体识别则不仅需要识别实体的边界，还需要识别实体的类型。因为这三项任务不同，所以目前的技术较难采用单一模型处理全部问题。

机器学习中的“没有免费的午餐”定理指出，必须设法寻找更适合当前任务的语言模型，而“丑小鸭”定理指出，必须寻找适合当前任务的有效特征，二者恰好都强调了先验知识的重要性。从已有的技术角度来看，倾向于运用更有效的模型解决特定的任务，再有机地结合各项处理结果；从信息增益角度来看，多种知识源分析的方法也正设法充分地利用先验知识，来提高词法分析中各个子任务的性能。

基于以下三种观点设计一种从处理流程上作适当优化的一体化词法分析系统：①分词、词性标注、命名实体识别之间的协调处理能够改善整个词法分析系统的性能；②采用易于融合更多统计特征与语言知识的模型有助于改善词法分析系统的性能；③恰当的特征集（如增加远距离特征）有助于改善词法分析系统的性能。也就是说，只有当系统能够较好地描述词法知识时，才能获得好的词法分析性能[1]。基于以上三种观点，从易于利用领域知识以及构建实用化词法分析系统的角度出发，采用各个子任务协作处理的方法构建实用的中文词法分析系统（本书称为 ELUS 词法分析系统），如图 1.1 所示。

图 1.1 中，基本分词模块完成词典词切分、仿词识别与派生词识别以及新词发现的任务，同时识别出仿词与派生词的类型。评测实验表明基本分词模块的精确率和召回率指标性能约为 98%[①]，而基本词性标注在不考虑未知词与复杂虚词时，可获得约 97%的标注精确率。前两步的处理结果为命名实体识别提供较为准确的词信息与词性信息。反过来，在分词与词性标注过程出现的未知词中，命名实体占主要部分，相比来说它更难处理。而词特征与词性特征会有助于命名实体的识别。尽管如此，不能忽略前续操作中的错误切分带来的影响，例如，“孙/桂/平等”中错误切分“平等”，从而易使识别过程无法复原实体，不过这样的一些歧义问题都将在歧义边界判别模块中得以修正。

复杂歧义可在前续处理后利用更加高级的特征进行消解[2]（消歧部分阐述见 4.6 节），如远距离约束“只有→才能”用于消歧“才能”或“才/能”，所以在精确分词模块主要针对这类复杂歧义进行消歧处理。在完善分词之后，词性标注模

① 本章采用北京大学的分词、词性标注、命名实体的定义标准。

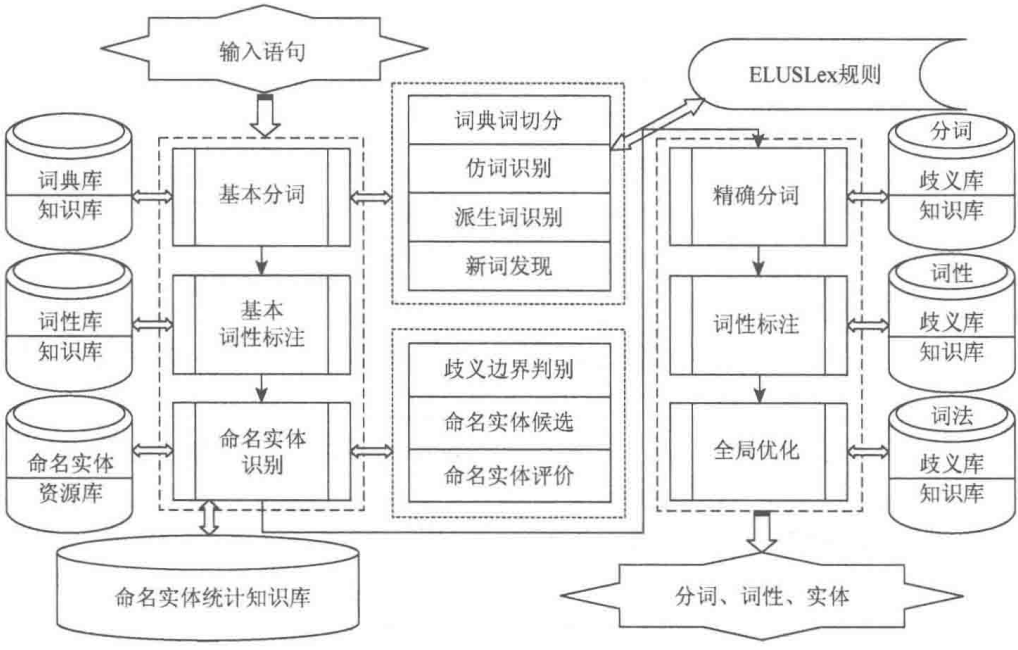


图 1.1　ELUS 词法分析系统的体系结构

块需对重切分句子重新标注词性；用命名实体识别结果标注词性；用消歧模型对复杂兼类词进行消歧。

## 1.2　词典与基于规则分词

### 1.2.1　快速索引词典

分词方法可分为有词典分词和无词典分词。有词典分词是指分词系统在分词时利用系统内预先收集存储的词，如词典内包括高校、生活、丰富多彩，因此有助于对“高校生活丰富多彩”语句快速分词。无词典分词是指分词系统构建时没有预先收集存储的词。无词典分词一般是在大规模文本形成的语料库上统计各字之间的紧密程度，按照构成词的多个字之间结合的紧密度、词与周围文字搭配相对松散的语言现象，构建统计评价函数，度量结合更为紧密的固定搭配字串。为了能对新句子进行分词，无词典分词系统也需要收集固定搭配的字串，并构建词典，只不过该词典是系统从无分词的文本语料库中依照统计分析自动收集的。相比较，有词典分词系统中的词典是分词系统构建时就预先给定的或者是从有分词的文本语料库中收集，并通过人工精心筛选而形成的词典。考虑到语言现象的复杂性，一般来说，有词典分词系统性能更优。

在系统实现上，无论有词典分词还是无词典分词都构建词典数据结构用于存储词信息。这里称为词典是因为分词往往与后续词性标注等自然语言处理过程结合，所以一个词可能还标注候选词性、拼音或者其他语言处理上所需要的信息数据。将所有词及其相关信息集成在一起构成词典。

一个简单的词典可以使用列表实现，如线性表、哈希表（Hash table），但需要考虑词典的查找效率，如果用线性表实现，单纯的线性逐一查找效率较低，通常至少需要配合使用二分查找技术，即将所有词按照从小到大排序，然后使用二分查找（折半查找）技术。哈希表存储是另一种典型的常见索引查找技术，其效率与哈希函数的映射效率密切相关，但通常哈希表构建的词典查找效率高于二分查找技术。

高效的词典索引结构能够有效地提高分词的速度。考虑到分词时需要快速判别某一个候选是否是词，并考虑一个汉字可能与多个字构成词，而候选词判别的过程又恰是逐一汉字匹配的过程，因此可以采用树形结构构建词典。其根是一个单一的字，而以该字为前缀的所有词则形成树形结构，图 1.2 表示以“社”为前缀的几个词构成的树形结构。

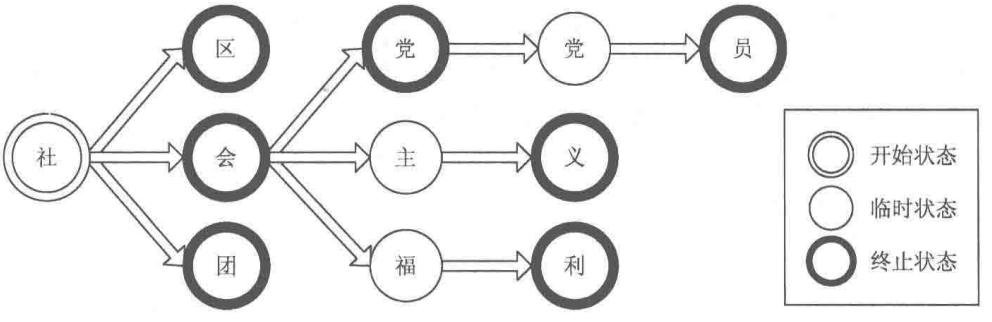


图 1.2　树形结构的词典构成举例

字典树又称作 Trie 树、单词查找树或者前缀树，是一种用于快速检索的多叉树结构，其插入、查找的时间复杂度均为 $O(N)$，其中 $N$ 为字符串长度。每个汉字都可以形成一个 Trie 树，这样整个词典是众多 Trie 树构成的集合。参考图 1.2，在树形结构中，词典词起始于“开始状态”（start state），结束于“终止状态”（end state）。在给定句子进行候选词匹配运算时，可以按照编译原理中的超长匹配方法，把经过的每一个状态均构成一个候选词，于是，当采用基于树形结构的词典时，利用词典生成词候选的过程则如同有限状态自动机识别的过程。

通过 Trie 树，以任何字开始进行词候选搜索的过程就变为与树各个分支匹配的过程，当到达一个终节点时就形成一个候选词。因此在分词中存在两

种典型的应用方式：一是寻找最长匹配词；二是找出匹配过程中途经的所有候选词。

### 1.2.2　正向最大匹配分词

正向最大匹配分词属于有词典的基于规则分词算法，它是从指定字开始寻找最长匹配词，当找到最长匹配词之后，再接着寻找下一个最长匹配词，依次循环，直到找到最后一个词，完成整个句子的分词工作。例如，“他们是社会党党员”，按照正向最大匹配分词算法，首先从“他”字开始，查找词典的最长匹配词是“他们”，再从“是”字开始，只有“是”作为一个词，然后从“社”字开始，参考图1.2，向后逐字匹配，找到最长匹配词“社会党党员”。于是分词的结果为“他们/是/社会党党员”。本例中，在进行“社”字开始的最长词匹配中，虽然途经“社会”“社会党”，但是仍然没有“社会党党员”字串长，所以按照最大匹配原则，长匹配字串优先，可将“社会党党员”划分为一个词。

正向最大匹配分词的过程就是从句子开头去匹配最长词，然后从接下来的字去匹配最长词，依次寻找，找到全部词。分词中需要假设单个字也是词，因此对于在词典中不存在词的字，假设这个字本身就是一个词。

单纯的正向最大匹配分词方法具有较高的性能，虽然没有统计分词性能高，但通常也能满足较多文本分析与文本挖掘任务的需求。但就分词技术本身来看，该方法主要面临两个问题。第一个问题是有些词切分得不准确，例如，“市场中国有企业才能发展”切分为“市场/中国/有/企业/才能/发展”，显然，出现了“中国/有”切分错误，“才能”应该是“才/能”。这类错误通常需要结合上下文环境进行分词歧义消解。第二个问题是词典中没有的词无法按照词划分出来，词典中没有的词称作未登录词或称作未知词。未登录词识别是各种分词方法所面临的一个重要问题。对于未登录词再进一步划分，可分为仿词（factoid word）、命名实体（named entity）和新词（new word）。对于仿词可以采用1.3节的处理方法来识别，对于命名实体可以采用第3章的方法进一步处理，而对于新词的识别可以通过新词识别技术以及基于文本上下文环境下的方差评价的方法进行处理。

### 1.2.3　反向最大匹配分词

反向最大匹配分词属于基于规则的分词算法，它对一个句子从后向前进行最长词匹配，逐一确定每个最长匹配词。可见反向最大匹配分词是从句子的最后一个字，向句首方向逐一匹配每个词，而正向最大匹配分词是从句首第一个字，向

句尾方向逐一匹配每个词。因为反向最大匹配分词是从句子末尾向前匹配，所以相当于对字串进行反向查询，如图 1.2 所示，查询“义主会社”，为此应用在反向最大匹配分词的词典需要构造与图 1.2 相反的索引，这里称作反向词典，如“义主会社”，实验表明该方法相比二分查找技术和哈希表存储有效地提高了分词的效率。如果存在正向词典，很容易编制程序实现字串反转，然后构造反向词典。

对“他们是社会党党员”进行分词，首先利用“员”查找最长匹配词，找到“社会党党员”，再从“是”查找，单独成词，然后从“们”查找，找到“他们”。最后的分词结果为“他们/是/社会党党员”。

大规模分词数据实验评价发现，反向最大匹配分词的切分精确率要比正向最大匹配分词的切分精确率高些。例如，对于“市场中国有企业才能发展”切分为“市场/中/国有/企业/才能/发展”，其中“中/国有”切分正确，而“才能”切分错误，应该是“才/能”。当都采用树形快速索引词典时，词典的插入、查找的时间复杂度均为 $O(N)$，分词速度相同，都属于基于规则的快速分词方法。

正如正向最大匹配分词所面临的困难，反向最大匹配分词也面临着切分歧义（segment ambiguous）和未登录词识别（unknown word identification）问题。这两个问题的典型解决方案也是切分歧义消解方法和仿词识别、命名实体识别以及新词识别技术。

## 1.3　仿词识别与最少分词技术

### 1.3.1　基于自动机的仿词识别

仿词主要包括数值词、日期词、时间词等，可以识别的仿词类别如表 1.1 所示。按 1998 年上半年《人民日报》语料库中统计，数值词的分布占语料库词分布的 3.71%，日期词和时间词占语料库词分布的 1.75%。在词法分析中，识别这类词是非常重要的：①仿词变化形式多样，属于未登录词的重要部分；②同一类仿词具有相似作用，识别的意义不仅体现在识别这类词的本身，还可以在语言模型的统计中将其视为一类，从而提高模型的处理能力；③仿词又可看作命名实体识别的一部分，识别不同的仿词还可以为后续语言处理（如句法分析）或直接应用（如自动文摘）提供基础。

**表 1.1　仿词类别**

| 仿词类别 | 包含的词类型 | 举例 |
| --- | --- | --- |
| Number | integer，percent，real 等 | 2910，46.12%，零点五，20.542 |
| Date | date | 2004 年 5 月 12 日，2010-05-03 |

续表

| 仿词类别 | 包含的词类型 | 举例 |
| --- | --- | --- |
| Time | time | 5：15，十点二十分，晚上 6 点 |
| English | English word | Hello，How，are，you |
| www | website，IP address | http: //www.jiangw.cn；192.168.140.133 |
| Email | Email | jiangw@hit.edu.cn |
| Phone | phone，fax | +86-451-86412114；（0451）86412114 |

仿词可以利用正则表达式（regular expression）来表示，因而可以利用有限状态自动机（finite state automaton，FSA）识别。当给定一个输入符号（input symbol）和当前状态（current state）时，确定性有限状态自动机（deterministic FSA，DFA）仅有唯一的下一个状态（next state），因而它是非常有效的。然而，人们更习惯于书写非确定性有限状态自动机（non-deterministic FSA，NFA）规则。NFA 允许几个下一个状态对应给定一个输入符号和当前状态。每一个 NFA 有一个等价的 DFA，于是借鉴自动机的方法是制作一个编译器（称为 ELUSLex），用 ELUSLex 将 ELUSLex 元规则（表 1.2）编译为一个 DFA。

**表 1.2　ELUSLex 元规则描述方式举例**

| |
| --- |
| <digit>->[0..9]\|[0..9]；//define Arabic numerals |
| <integer>:: = {<digit> + }；//define Arabic Integer |
| <real>:: = <integer>(.\|．\|·\|点)<integer>；//define float |
| <day>-><integer>日；//define day |
| <month>-><integer>月；//define month |
| <year>-><digit><integer>年；//define year |
| <date>:: = <year><month><day>；//define date |

ELUSLex 主要用于识别仿词，所以表 1.2 中的 ELUSLex 脚本元规则并非用于产生语言，而是用于识别关键词，例如，虽然“<month>-><integer>月”可以识别“13 月”，但现实文本很少出现这种情况，此外也可以通过“<month>-> [1..9]月|[1..9]月丨1[0..2]月|1[0..2]月”来定义更符合“1 月到 12 月”的 ELUSLex 脚本。本书定义的 ELUSLex 编译器从以下三方面增强规则的描述能力。

（1）允许的元规则描述：<Non-terminator>，terminator，{Loop block}，{Loop block + }，{Loop block*}，[Range block]（e.g. [a..z]、["a".."z"]），|，（Optional block），（Optional block + ），（Optional block *）。

（2）转义表达：元规则中的符号在表示终结符时，可以使用双引号括起来的方式来表达，如“（”“|”“）”。

（3）产生式类型：“->”用于表示临时规则，不被识别。临时规则便于后续规则的描述。“::=”定义可识别的规则，是识别仿词时使用的规则。

ELUSLex 元规则方式存在如下优点：①易于表示多种需识别的仿词类型；②便于定义、识别不同仿词类型，如问答系统中需要详细定义各种电话的识别规则；③可以根据实际的识别需要，通过简单地修改规则来完成不同仿词的定义。例如，在 Sighan 2005 评测中，ELUSLex 编译器方便地实现不同的仿词定义标准，包括北京大学、微软亚洲研究院、香港城市大学和台湾“中央研究院”标准。

### 1.3.2　仿词识别规则举例

表 1.3 给出了北京大学语料库中的仿词对应的 ELUSLex 脚本元规则。

**表 1.3　ELUSLex 脚本元规则举例**

| 规则 | 举例 |
|---|---|
| 基本规则 | <quan_jiao_digit->[０…９]；<br><ban_jiao_digit->[0..9]；<br><chinese_digit->零\|一\|二\|三\|四\|五\|六\|七\|八\|九\|十\|〇；<br><十百千万>->十\|百\|千\|万\|亿\|十万\|百万\|千万\|百亿\|千亿\|万亿；<br><letter_lower->[a..z]\|[ａ…ｚ]；<br><letter_upper->[A..Z]\|[Ａ…Ｚ]；<br><digit-><ban_jiao_digit>\|<quan_jiao_digit>；<br><letter-><letter_lower>\|<letter_upper>；<br><cn_integer->{<chinese_digit><十百千万>（零*）*}{<chinese_digit>+}；<br><en_integer->{<digit>+}；<br><en_real_or_integer-><en_integer>\|<en_integer>（.\|．\|•\|点）<en_integer>；<br><base_integer-><en_integer>\|<cn_integer>； |
| 数值规则 | <real>::=<en_real_or_integer>（+\|-\|*\|/\|十\|一\|＊\|/\|×）<en_real_or_integer>；<br><integer>::=<base_integer>；<br><real>::=<en_integer>（.\|．\|•）<en_integer>\|<cn_integer>（．\|•\|点）<cn_integer>；<br><real>::=<integer>分之<integer>\|<integer>分之<chinese_digit>；<br><real>::=<real>（%\|%\|‰）；<br><real>::=<integer>（%\|%\|‰）；<br><integer>::=<base_integer>几；<br><real>::=<real><十百千万>；<br><real>::=（百\|千）分之（<real>\|<integer>）；<br><orderinteger>::=第<integer>\|第"（"<integer>"）"；<br><integer>::=几（十*）<十百千万>；<br><integer>::=两（<十百千万>）（<base_integer>*）；<br><integer>::=<base_integer><十百千万>； |
| 短整型 | <short_digit->\|<digit>\|<digit><digit>；<br><short_chinese_digit->\|<chinese_digit>\|<chinese_digit><chinese_digit>\|<chinese_digit><chinese_digit><chinese_digit>；<br><short_integer->\|<short_digit>\|<short_chinese_digit>； |

续表

| 规则 | 举例 |
| --- | --- |
| 时间日期 | <月 suffix>->份；<br><日>-><short_integer>日；<br><月>-><short_integer>月（<月 suffix>*）；<br><年>-><digit>{<digit>+}年；<br><年>-><chinese_digit>{<chinese_digit>+}年；<br><date>:: = <年>|<月>|<日>；<br><time>:: = <short_integer>:<short_integer>:<short_integer>；<br><time>:: = <short_integer>（时|点|点钟）；<br><time>:: = <short_integer>分；<br><time>:: = <short_integer>（秒）； |
| 其他识别规则 | <bili>:: = <en_real_or_integer>（：|:|:）<en_real_or_integer>；<br><englishword>:: = <letter>{<letter>|<digit>|—|/*}；<br><IP>:: = <integer>.<integer>.<integer>.<integer>；<br><www>:: = http: //<englishword>{.<englishword>}； |

需要说明的是：①元规则按从前到后嵌套使用；语法中保留的特殊字符，如果作为终结符，必须使用双引号括起来。②允许元规则语法结构：＜非终结符＞、终结符、{循环块}、{循环+}、{循环*}、[a..z]、["a".."z"]、|、(括号块)、(括号+)、(括号*)。③利用“->”定义临时产生式，其不被识别；利用“:: =”定义需识别的产生式。

### 1.3.3　基于词网格的最少分词技术

词网格（word lattices）是一种形象地描述一个待分词语句与其所形成的全部候选词构成的路径的方法。图 1.3 和图 1.4 是两个句子对应的词网格，最上面一行代表单个字构成的词形成的一个路径，下面对应着多个字串形成的词候选。在依据给定的句子构建词网格时，快速索引词典有助于高效率判别指定字连续字串是否是候选词，从而快速生成词网格。

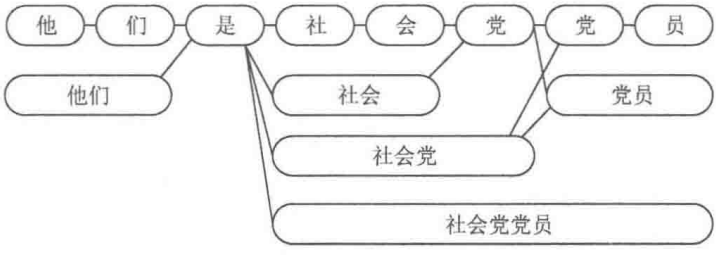


图 1.3　词网格举例