电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

# 硕士学位论文

**MASTER THESIS**


论文题目 中文开放式实体关系抽取研究与实现

| | |
|---|---|
| 学 科 专 业 | 计算机科学与技术 |
| 学 号 | 201321060646 |
| 作 者 姓 名 | 李 杨 |
| 指 导 教 师 | 耿 技 教授 |

分类号 _____ 密级 _____

UDC [注1] _____

# 学 位 论 文

## 中文开放式实体关系抽取研究与实现

（题名和副题名）


李 杨

（作者姓名）

指导教师 耿 技 教 授

电子科技大学 成 都

（姓名、职称、单位名称）


申请学位级别 硕士 学科专业 计算机科学与技术

提交论文日期 2016.03.18 论文答辩日期 2016.04.15

学位授予单位和日期 电子科技大学 2016 年 6 月

答辩委员会主席 _____

评阅人 _____

注 1：注明《国际十进分类法 UDC》的类号。

# Research and Implementation of Chinese Open Entity Relation Extraction

A Master Thesis Submitted to

University of Electronic Science and Technology of China


Major: **Computer Science and Technology**

Author: **Yang Li**

Advisor: **Professor Ji Geng**

School : **School of Information and Software Engineering**

## 独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：_____ 日期：　　年　　月　　日

## 论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：_____ 导师签名：_____

日期：　　年　　月　　日

# 摘　要

随着互联网信息不断增长，传统的返回检索页面的信息检索方式已经难以满足用户全面快速获取信息和知识资源的需求。实体关系抽取作为信息抽取重要组成部分，自动化地从自然语言中抽取实体关系元组的结构化信息，从而为用户提供更加智能的信息检索方式，帮助用户快速理解、掌握互联网中日益增长的信息。

传统的关系抽取主要针对某些特定领域而且需要预先定义关系类型，因此这种方法可移植性差，难以处理大规模网络文本数据。为从网络文本中抽取实体关系元组，本文提出了一种中文开放式实体关系抽取算法 DPM：首先，通过自动获取大量高质量实体关系元组和其回标的语句作为训练语料，并通过训练语料抽取得到基于词性和依存句法分析角色的关系模式；其次，为提高预处理的准确性，本文使用依存句法分析统计规律对待抽取文本进行兼类词处理；再次，使用第一步中学习得到的关系模式对兼类词处理后的语料进行模式匹配并抽取候选实体关系元组；最后，对抽取的候选实体关系元组使用机器学习方法进行置信度评估，得到其中高置信度的实体关系元组。通过在公开数据集 Wiki-500、Sina-500 以及本文构建的 Tecent-500、Simple-500 数据集上对 DPM 算法进行验证。实验结果中，本文提出的中文开放式实体关系抽取算法 DPM 的 P-R 曲线基本位于相关工作的右上方。这表明 DPM 算法的有效性，即当大多数情况下准确率相同时 DPM 算法的召回率更高，当召回率相同时 DPM 算法的准确率更高。

此外，本文实现了一个中文关系抽取原型系统，系统一方面利用百科网页中结构化和半结构化信息抽取实体上下位关系以及属性关系，同时为进一步丰富关系知识库中的实体关系元组，系统使用本文提出的开放式实体关系抽取算法DPM从纯文本中抽取实体关系元组。

**关键词：** 信息抽取，关系抽取，实体抽取，知识库

# ABSTRACT

With the increasing of information on the Internet, traditional search method is difficult to satisfy the needs of users who want to master the information and knowledge resources quickly and fully. As a vital part of information extraction, entity relation extraction strives to extract relations and attributes of entities from the semi-structured and unstructured natural language automatically. It helps users to understand the increasing network information more efficiently and provides more intelligent information retrieval services for users.

Traditional information extraction is often limited by only extracting predefined relation types so the portability of the method is poor and the method is difficult to deal with large-scale web data. In order to extract relation tuples from web data, this thesis proposes an open relation extraction method DPM. First, the method automatically obtains the training corpus that consists of large amounts of high quality relation tuples and their corresponding sentences. Then it learns numerous relation patterns that encode with dependency parsing roles and parts of speech from large corpus. Second, in order to improve the quality of pre-processing, this thesis deals with trans-classed words by the dependency parsing statistical laws. Third, we use learned patterns in the first step to extract candidate relation tuples. Finally, we evaluate the quality of relation tuples by using logistic regression to obtain high quality relation tuples. We use open dataset Wiki-500、Sina-500、Tecent-500 as well as Simple-500 to prove the validity of DPM. The P-R curve of DPM is almost in the upper right of related work's. The experiments show the effectiveness of the DPM algorithm, in other words our study demonstrated that DPM performs well on recall when the accuracy is at the same level; and that When the recall is at the same level, DPM performs well on accuracy.

In addition, we proposed a relation extraction system which can not only extract hyponymy relations and attribute relations from Baidu Encyclopedia but also can use DPM method to extract relation tuples from web data to population relation knowledge base.

**Keywords:** information extraction,relation extraction,entity extraction,knowledge base

# 目 录

**第一章 绪 论**.....1

1.1 研究背景及意义.....1

1.1.1 研究背景.....1

1.1.2 研究意义.....1

1.2 国内外研究现状.....3

1.2.1 实体抽取.....3

1.2.2 实体关系抽取.....4

1.3 论文的研究内容.....6

1.4 本文的层次结构安排.....7

**第二章 相关技术背景及算法**.....8

2.1 中文分词.....8

2.1.1 中文分词概述.....8

2.1.2 中文分词研究现状.....8

2.2 词性标注.....9

2.2.1 词性标注概述.....9

2.2.2 词性标注研究现状.....10

2.2.3 兼类词.....11

2.3 依存句法分析.....12

2.3.1 依存句法分析概述.....12

2.3.2 依存句法分析研究现状.....12

2.4 机器学习相关算法.....14

2.4.1 聚类.....14

2.4.1.1 聚类概述.....14

2.4.1.2 聚类研究现状.....15

2.4.1.3 K-means 算法.....16

2.4.2 Logistic 回归.....17

2.5 本章小结.....20

**第三章 中文开放式实体关系抽取算法研究**.....21

3.1 关系抽取算法流程.....21

3.2 关系模式抽取.....21
3.2.1 训练语料的获取.....22
3.2.2 关系模式学习.....24
3.2.3 模式聚类.....24
3.3 候选关系抽取.....27
3.3.1 兼类词处理.....27
3.3.2 实体短语识别.....30
3.3.3 模式匹配和关系抽取.....31
3.3.4 关系扩展.....32
3.4 候选关系过滤.....33
3.5 本章小结.....35
**第四章 中文开放式实体关系抽取算法结果分析**.....36
4.1 数据集.....36
4.2 评判标准.....36
4.3 关系抽取对比.....37
4.3.1 ZORE 算法对比分析.....37
4.3.2 DPM-NoTCW 算法对比分析.....39
4.4 模式评估.....40
4.5 总结.....41
**第五章 中文实体关系抽取原型系统实现**.....42
5.1 系统框架.....42
5.1.1 系统功能模块.....42
5.1.2 系统流程.....43
5.2 系统开发环境及相关工具.....44
5.2.1 系统开发环境.....44
5.2.2 系统使用的相关工具.....45
5.3 数据爬取和解析模块.....47
5.3.1 数据爬取和解析.....47
5.3.2 重要数据结构和功能函数.....48
5.3.3 输入输出.....48
5.4 文本预处理模块.....48
5.4.1 自然语言处理.....49

5.4.2 停用词处理.....49
5.4.3 重要数据结构和功能函数.....49
5.4.4 输入输出.....50
5.5 百科网页关系抽取模块.....50
5.5.1 上下位关系抽取.....50
5.5.2 属性关系抽取.....51
5.5.3 重要数据结构和功能函数.....53
5.5.4 输入输出.....54
5.6 开放式关系抽取模块.....54
5.6.1 开放式关系抽取.....54
5.6.2 重要数据结构和功能函数.....55
5.6.3 输入输出.....56
5.7 系统测试和关系知识库展示.....56
5.7.1 词性标注测试.....56
5.7.2 依存分析测试.....57
5.7.3 百科属性关系抽取测试.....58
5.7.4 开放式关系抽取测试.....58
5.7.5 关系知识库展示.....59
5.8 总结.....60
**第六章 结束语**.....61
6.1 全文总结.....61
6.2 不足与下一步展望.....62
**致 谢**.....63
**参考文献**.....64
**攻硕期间取得的研究成果**.....68

# 第一章 绪 论

本章主要描述了移动互联网的普及和快速发展趋势之下关系抽取课题研究的背景和意义，说明了关系抽取为用户在海量的信息资源中快速获取符合用户需求的信息和知识库构建方面起着重要作用。同时对国内外的研究现状和本文的研究内容、组织结构做简要的介绍。

## 1.1 研究背景及意义

### 1.1.1 研究背景

互联网技术从网页链接发展到数据链接，表明了Web技术正在逐渐向语义网络演变，语义网络不同于网页链接其是由数据构成的网络[1]。这种技术可以为使用者提供更加快速、方便的查询环境，其可以为使用者直接返回经过加工、推理后得到的答案和知识。

传统的信息检索技术主要依据用户的查询，返回经过排序的网页列表。由于返回的大量网页列表需要人为的进行筛选，因此这种信息检索方式难以满足用户快速、准确获取相关答案和信息的需求。随着网络中数据量不断的增长，这种检索方式已经难以满足用户掌控信息和知识资源的需求，比如从互联网中直接获取相关问题的答案而无需花费大量的时间在返回的结果中人工定位答案等。因此，信息抽取技术的出现，为解决信息检索问题提供了新的思路，得到了学术界和业界普遍的关注。信息抽取目的是通过自然语言处理等相关技术将非结构化的纯文本转化为结构化的文本数据。其中最主要的是从纯文本中识别并抽取出实体以及实体关系元组等，然后将抽取结果进行结构化的存储，如存储在关系数据库、图数据库中，为数据的分析和查询相关应用提供数据基础。

实体关系抽取作为信息抽取一项重要的子任务主要是为了识别并抽取实体间的关系。ACE和MUC两个评测会议促进了关系抽取研究的迅速发展。经过多年的不断改进和发展，信息抽取技术也取得了重要进步和改善：从特定格式数据转向无结构化纯文本数据，从特定领域转向开发领域。

### 1.1.2 研究意义

实体关系抽取涉及机器学习和自然语言处理等技术，具有重要理论和实用意义。通过关系抽取，可以将网络中海量的信息转化为符合人类认知世界的表示形

式，进而为用户提供一种更好地组织、管理以及如何利用这些信息的方法。现如今，实体关系抽取可以应用于知识库的构建（如YAGO），智能语义搜索（如搜狗知立方）以及深度问答(如Watson)等。

(1) 知识库的自动构建

目前的知识库内容主要来自于人工输入或业务结构化数据，然而这些仅占人类知识的很小一部分。还有大量的实体关系元组数据以自然语言的形式隐式的存储在非结构化网络海量数据中。但由于其是以无结构化的形式散落在互联网数据中，

表 1-1 相关知识库

| 知识库 | 产品 | 数据来源 |
|---|---|---|
| Knowledge Graph<br>Knowledge Vault | Google 搜索引擎<br>Google Now | 维基百科<br>Freebase 纪实年鉴<br>网页公开数据 |
| Freebase | Metaweb 语义搜索引擎 | 维基百科 |
| Satori<br>Probase | 微软 Bing 搜索引擎<br>微软 Cortana | Wolfram Alpha<br>网页公开数据 |
| 人立方知识库 | 微软人立方 | 网页公开数据 |
| Watson 知识库 | Watson 自动问答系统 | 各种网络辞典文集<br>世界图书百科全书 |
| 知立方 | 搜狗搜索引擎 | 网页公开数据 |
| 知心知识库 | 百度知心 | 网页公开数据 |
| Xlore 知识库 | Xlore 知识图谱 | 中英文的百科数据 |
| YAGO 知识库 | YAGO | 维基百科 |
| Zhishi.me 知识库 | Zhshi.me | 中文百科数据 |
| NELL 知识库 | NELL | 网页公开数据 |

难以直接被直接利用。因此信息抽取技术和关系抽取技术得到关注和发展。如上表 1-1 给出了当前主流知识库及其相关的应用，其中WolframAlpha是包含信息数最多的知识库，Probase是当前包含概念数量最多的知识库 [2]。

(2) 智能语义搜索

在该应用中，当使用者进行查询时，首先系统会借助信息抽取和知识库对用户查询的关键字进行相关的解析，进而将该关键字中的实体映射到知识库中对应的相关概念上；然后依据知识库中的内容返回结构化的知识，如谷歌搜索引擎页

面返回的知识卡片。智能语义搜索在国内处于业界领先水平的是搜狗公司，其产品知立方可以基于语义网推理补充用户查询的相关信息。

(3) 深度问答

在该应用中，系统首先借助知识库对用户查询的问题进行句法和语义分析，并将其转化为结构化的查询形式（三元组查询形式），然后直接在知识库中进行查询得到所要的答案。系统通常会对转化后的结构化查询语句进行多次等价的变换。如用户查询语句为：“奥巴马妻子的年龄是多少？”，则该查询可能转化为“米歇尔.拉沃恩.奥巴马的年龄是多少？”，然后再继续进行推理变换得到(米歇尔.拉沃恩.奥巴马，年龄，？)，最后在知识库中查询得到相应的答案。由于知识库中知识的有限性以及日常生活中新知识不断的产生，经常导致所查询的问题在当前的知识库中没有直接的答案，对此可以采用推理的技术从知识库中已有的相关信息得到目的答案。

传统的关系抽取方法作为关系抽取的初级方法，其需要预先定义关系类型，例如“朋友关系”，“同学关系”等等，同时构造大规模的训练语料，然后在特定领域的文本数据中识别实体间预先定义的实体关系类型。这很大程度上限制了信息抽取技术的发展和应用。面对网络海量的文本数据，如果仅仅在特定领域语料中抽取某些预定义的特定类别的关系，则可以抽取的关系元组数量有限。传统的关系抽取在实际的使用中面临多种问题和挑战，开放式实体关系抽取作为一种改善方式受到研究者的关注。由于网络海量数据以及人们对获取知识的更高诉求，开放式关系抽取具有如下重要的意义:第一点，该方法不需预先定义关系类型，能够抽取到除定义的关系类型之外更多的关系;第二点，不限定待抽取数据的领域，可以抽取各种领域的文本数据包括领域开放的网络文本数据;第三点，开放式实体关系抽取无需依赖标注数据，大大的减少人工标注语料的工作量。

## 1.2 国内外研究现状

### 1.2.1 实体抽取

实体抽取，是指从文本数据集中自动识别和抽取出表示实体概念的词汇。实体抽取是关系抽取的前提，其抽取的质量直接影响后续关系抽取的效果。

最初始时实体抽取任务主要是识别特定领域数据语料中的人名、地名等具有实际含义的信息[3]。Rau等人[4] 采用预先人为定义和编写规则的方法识别公司类实体。然而由于制定规则时需要专家耗费大量的精力，此外这些制定的规则其迁移性也相对较差，对于不同领域的数据难以适用。为了应对基于规则的实体识别方

法的局限性，研究者开始尝试采用机器学习方法解决此类问题。如Liu等人[5]使用KNN（K近邻算法）和CRF（条件随机场模型）从文本数据中识别和抽取出大量的实体。由于监督学习算法受到训练语料规模的限制，进而影响了该任务的性能。因此可以结合一些预先定义的规则抽取实体。如Liu等人[6]采用监督学习方法和规则相结合的方法来识别实体，具体使用Maximum Entropy Model（最大熵算法），最终本文提出的方法准确率达到 0.727，召回率达到 0.715，F值达到 0.721。

随着该技术的不断发展和进步，面向特定领域的实体抽取已经难以满足需求，开放的网络数据得到越来越多关注。因此，传统的实体分类体系已经难以适应现在的工作，需要建立一个全面的分类体系。如Sekine等人[7]2002 年通过总结归纳将实体分为 150 个类别。为进一步提高分类的合理性，Ling等人[8]根据Freebase知识库使用的实体分类层次结构进行进一步的处理，最终得到 112 种类别。

然而，面对来自互联网不断增长的海量数据，其需要的命名实体的类别更多、更细。基于此预先定义命名实体分类体系然后抽取实体的方法已经难以满足实体识别任务的需求。开放式命名实体识别技术不需要限定领域和语料类别可以有效避免对实体分类体系的构造。这种方法主要是给定某一特定类别的种子实体，然后从给定的文本数据中抽取得到上下文环境相似的命名实体[9]。开放式实体识别技术面临的主要挑战是如何从给定的有限的种子实体集合中自动的从文本数据中抽取得到具有明显区分能力的模式。如Whitelaw等人[10]在海量冗余网络数据中首先识别其中明显的模式，然后利用得到的模型识别新的命名实体，接着再利用这些新识别的命名实体去发现更多的模式，迭代的抽取命名实体。

### 1.2.2 实体关系抽取

经过实体识别之后可以得到大量相互独立的实体列表，然而这些相互独立的实体只能反映出文本中包含哪些实体，不能反映出文本中蕴含的实体之间的关系。因此，为了得到文本数据中蕴含的语义关联信息，还需要从文本数据中抽取得到实体与实体间的关系。

关系抽取最初主要采用人工预先定义规则的方法，然后使用这些预先定义的规则匹配待处理语句进而抽取实体间的关系。该方法虽然简单但是具有如下不足之处：首先，对制定规则的人要求较高，需要其对特定领域的知识具有较深的积累；其次，预先制定关系规则需要花费专家大量的时间；最后，由于不同领域文本数据的风格不同，因此这些预先定义的规则扩展性不强，难以应用于其他领域文本数据的关系抽取。由于上述基于规则的实体关系抽取方法的不足之处，研究者通过使用统计的方法抽取实体之间的关系。如Kambhatla等人[11]使用最大熵算法

来抽取实体之间的关系。

随后，这种基于统计机器学习的关系抽取方法得到快速的发展。如Bunescu等人[12]使用最短依存树核函数识别实体之间关系，其结果明显优于当时其他的关系抽取系统。由于有监督学习方法的准确性是建立在大量训练语料的基础上的，然而训练语料的获取需要花费大量的人力。因此对训练语料需要相对较少的半监督和无监督的机器学习方法更具有研究意义。Zhang等人[13]利用无监督机器学习方法对实体之间多种类型的关系进行抽取，且该方法取得了较好的性能。

以上所有的传统关系抽取方法都需要提前定义好关系的类别，如朋友关系、夫妻关系和师生关系等。然而在面向海量互联网数据中，预先定义出所有实体关系类型是十分艰巨的同时也是不可能的。为了避免预先定义各种实体关系类型，面向开放域的信息抽取方法随之被提出。华盛顿大学Banko等人[14]于 2007 年首次提出面向开放域的信息抽取方法并实现了第一个开放式关系抽取系统TextRunner，其性能优于当时其他关系抽取方法。

面向开放域的关系抽取直接使用语句中的关系指示词作为候选关系，不需要像传统关系抽取那样预先定义关系类型，下表 1-2 展示了两种关系抽取方法的区别。由于开放式关系抽取的这一优点，关系抽取取得不断的进步和性能的改善。继TextRunner系统之后，Wu等人[15]发布了WOE关系抽取系统，其基于百科信息框中的属性信息自动回标语句构建训练语料，并且相对TextRunner系统其性能有提高。由于TextRunner和WOE系统抽取的关系元组中存在部分无意义以及不符合逻辑的关系元组，因此Fader等人[16]引入了语法限制以及字典约束来识别实体关系元组并提出了Reverb关系抽取系统。这进一步改进了关系抽取的性能。由于TextRunner、WOE和Reverb这些开放式关系抽取系统均无法识名词关系，因此Schmitz等人[17]提出了OLLIE系统，其可以抽取文本中的名词关系，进一步提高了关系抽取的性能。

表 1-2 关系抽取方法比较

| | 传统实体关系抽取 | 开放式实体关系抽取 |
|---|---|---|
| 输入 | 待抽取数据+预先标注数据 | 待抽取数据 |
| 关系 | 预先定义的关系类型 | 自动发现关系 |

鉴于传统的关系抽取方法和开放式关系抽取方法各自的优缺点，可以结合使用这两种方法。如Banko等人[18]基于条件随机场，结合实现了传统关系抽取方法和开放式关系抽取方法，其性能优于单独使用其中一种方法。此外，微软人立方项目中StatSnowball模型也同时结合了这两种关系抽取方法[19]。

现在开放式关系抽取的研究主要集中在英文，对其他的语言的研究相对较少，中文实体开放式关系抽取还处于起步阶段。现阶段中文开放式关系抽取系统主要有CORE和ZORE。CORE采用CKIP依存分析工具，分析语句的句法结构，然后使用依存分析规则抽取关系元组[20]。ZORE是现在最为先进的中文开放式关系抽取算法，使用语义模式和双向传播识别候选关系元组[21]。

当前关系抽取主要关注的二元关系，忽略了文本中存在的大量高阶多元关系。Alan 等人[22] 为了从文本数据中抽取多元关系实现了 KRAKEN 模型，使得关系抽取任务的召回率得以提高。其次，由于当前开放式实体关系抽取方法无法实现对语料中隐含的关系元组进行抽取。McCallum[23]使用联合推理的方法，实现了对语料中隐含实体关系的抽取。

## 1.3 论文的研究内容

本文在现有研究基础上，针对中文的特殊语法现象如兼类词问题，提出了改进的中文开放式实体关系抽取方法DPM。本文同时还实现了一个关系抽取系统。本文的主要工作如下：

(1) 自然语言预处理

对输入的文本进行自然语言处理，为使预处理结果达到相对较好的结果，本文通过调研多种自然语言处理工具如中科院的ICTCLAS，哈尔滨工业大学的LTP，斯坦福大学的自然语言处理工具，复旦的自然语言处理工具FudanNLP以及Zpar，本文最终采用Zpar对待处理文本进行分词、词性标注以及依存分析。

(2) 依存关系模式学习和聚类

利用大量关系元组，并结合其对应语句的自然语言处理结果自动学习大量高质量包含依存语义角色、词性的二元和多元关系模式。同时按照频率对学习的关系模式进行降序排序。此外在抽取中有些模式表示的实际含义相同，为提高模式的覆盖率和粒度，进一步对模式进行聚类以提高模式的覆盖率和粒度。

(3) 候选关系抽取

首先，通过兼类词词性规则对待抽取文本中的兼类词的词性进行调整，以纠正兼类词在不同语境下的词性。然后，选择学习得到的与待抽取语句匹配频率最高的关系模式抽取关系元组并对关系元组进行扩展和调整。

(4) 关系过滤

对抽取的关系元组使用Logistic分类器对其进行质量评估并过滤低质量的关系元组。Logistic过滤器中使用了 30 个特征，其中不仅包括浅层词法特征如句子的长

短，实体和关系词间的距离等，而且还包括如实体和关系词的依存语义角色等语义特征。

(5) 中文关系抽取系统的实现

本文实现的中文关系抽取系统首先利用百科网页抽取上下位关系、实体属性关系，此外为丰富关系知识库中的关系元组，系统使用开放式实体关系抽取算法DPM从纯文本中抽取关系元组。

## 1.4 本文的层次结构安排

本文研究内容和论文结构如下：

第一章绪论，主要阐述了实体关系抽取的背景和意义及其在国内外的研究现状，同时阐述了文章主要研究内容。

第二章相关技术背景和算法，主要介绍了和关系抽取联系紧密的背景知识和和相关的机器学习算法。

第三章中文开放式关系抽取算法研究，本章详细介绍了本文提出的开放式关系抽取算法 DPM。

第四章关系抽取系统，本章详细介绍了系统实现过程中使用的框架和技术，同时阐述系统中各功能模块的设计和实现。

第五章实验结果分析。通过多组对比实验，验证了本文提出的 DPM 算法的有效性和本文学习得到的关系模式的正确性。

第六章结束语，本章主要对文章的工作进行总结，指出了文章的不足以及待改进之处，同时提出了进一步的改进工作。

# 第二章 相关技术背景及算法

本文研究的开放式关系抽取是指从不限定领域的文本语料库（中文文本语料）中抽取出实体与实体间存在的不限定关系类别的关系元组。本章对在研究过程中主要使用的相关技术和算法进行介绍和说明。首先，介绍了本文对文本数据进行预处理使用的相关自然语言处理技术包括分词、词性标注以及依存分析；然后，介绍本文进行候选关系元组抽取时涉及的兼类词处理技术；最后，介绍本文在模式聚类环节使用的聚类技术，以及对抽取的候选关系元组进行过滤使用的 Logistic 机器学习算法。

## 2.1 中文分词

### 2.1.1 中文分词概述

分词是指将待处理文本中的语句依据一定的规则将其重新切分为一个个有意义的词序列的过程。然而中文不像英文语句，单词间以空格字符作为分界。中文语句则是由单独的汉字组成，然而具有语义的最小单位却是由汉字组成的词。平时生活中两个字组成的词汇使用的最多，同时也存在一些一个字的单字词以及多个字的专用词汇，因此中文分词相对英文分词难度更大。

### 2.1.2 中文分词研究现状

目前，主要存在如下三种分词方法：

(1) 基于字符串匹配的分词方法

在使用此方法进行分词，其中分词的字典和规则是必不可少的。该方法的主要过程是将准备进行分词的文本数据和分词字典做比对。若分词词典中存在某个字符串，那么认为比对成功，接着从文档中切分该字符串作为分词的结果，否则不予以切分。这种分词方法简单而且实用，但是由于其需要预先设定分词字典中词汇的内容，其难以将新出现的词汇加入字典中，因此随着新词的不断产生，分词的效果不断恶化。在分词的过程中，该分词方法存在两种分类方式：一种是按照待分词语句的扫描方向，可将该分词方法划分为正向匹配的分词方法和逆向匹配的分词方法；一种按照匹配长度可划分为最小匹配分词方法以及最大匹配分词方法 [24]。

(2) 基于规则的中文分词方法

本方法是指从语言学角度出发，利用计算机模拟人对语句的理解和分析的过程进而实现对语句的分词。1991 年，何克抗等人[25]首次使用专家系统对文本进行分词，该专家系统主要由知识库以及推理机组成的。由于知识库的构建工作量巨大而且异常困难，同时推理也涉及到相关复杂的人工智能技术，因此该系统难以广泛应用。2005 年张茂元等人提出了基于语境的中文分词方法，大大降低了系统的中文分词系统的复杂性[26]。

(3) 基于统计分词方法

该方法是建立在词是稳定的字符序列组合这一思想上的，因此在大规模统计语料中若干字符序列的组合同时相邻出现的频率越高，那么该字符序列的组合是一个词汇的可能性就更大。例如，可以统计语料中相邻出现的所以汉字组合的次数，然后计算得到每个组合中表示汉字间紧密程度的互信息。互信息值越大则该汉字组合构成词汇的可能性就越大。

基于统计的分词方法是通过建立分词统计模型，然后计算输出各种可能组成词汇的字符串的概率。该方法拟补了基于规则和字符串匹配分词方法对存在歧义词汇的识别以及未登录词汇识别的缺陷。该方法所应用的主要的统计模型有：N-gram、HMM以及Maximum Entropy Model等。为了充分利用多种分词方法的优点，在实际使用时我们通常将上文第二种和第三种方法结合使用，既利用了第一种方法高效率的特点，同时也利用了第二种方法在歧义词汇的识别和未登录词识别的优点。如王伟等人提出了基于 EM 算法的中文分词方法，其中重点分析解决了歧义词汇的分词问题，该方法在包含歧义词汇语料中分词的准确率达到85.36%[27]。2002 年中科院基于层叠隐马尔科夫模型实现了中文分词系统ICTCLAS，ICTCLAS的性能优秀，在学术界以及业界得到了广泛的推广和使用[28]。

## 2.2 词性标注

### 2.2.1 词性标注概述

词性是词的语法功能，表示词汇在一定的词类系统中的类别归属。文本中词汇的词性不但取决于已经选定的词的类别系统，同时也取决于词汇本身在语句上下文的语法语义功能。词性标注是指为文本中的所有的词汇标记一个在当前上下文中合适的词性标记，换句话说该任务需要确定每个词在文本中合适的词性（如名词、动词等）并标注出结果的过程。

汉语词汇在不同的上下文环境中词性往往也不相同，当某个词汇所在的上下文环境改变则该词汇的词性也会随之而发生变化，也就是说对于汉语来说同一个

词汇可以拥有多个不同的词性即为词性兼类现象。对于语句中出现词性兼类现象时，该语句的词性标注问题就更加困难。

### 2.2.2 词性标注研究现状

随着词性标注技术的研究和发展，目前词性标注技术大致分为如下三种，表2-1 对这三类方法进行了总结和比较。

(1) 基于规则的词性标注方法

该方法首先需要预先针对汉语的语法特点建立丰富的规则，然后使用预先定义的规则对待处理文本进行词性类别的标注。该方法具有如下缺点：第一点，对于预先获取大量具有区分度的规则需要耗费语言学家大量的时间和精力而且效率低下；第二点，在规则建立的过程中需要语言学家对语法知识有深入的认识和理解；第三点，这些预先定义的规则覆盖面有限，难以覆盖所有语言现象；第四点，人工预先定义的规则不一定都是可靠的其质量需要进一步的考察和验证；第五点，最后随着定义规则的数量不断增加，规则与规则之间可能不一致甚至彼此冲突，因此使用本方法时，应该协调好规则的覆盖率和准确度之间的关系。由于上述缺点使得这种方法难以得到广泛使用。例如美国布朗大学提出的词性标注系统TAGGIT其中包括 86 种词类标记，使用 3300 条规则对Brown语料库进行词性标注，系统的准确率达 77％[29]。由于基于规则的词性标注方法是利用人工去总结获取可靠的规则，其困难性巨大，所以Brill提出了一种可以自动的从语料中抽取出规则的方法[30]。因此，该方法获取的规则相对人工预先定义规则更可靠以及更适用。

(2) 基于统计学的词性标注方法

该方法首先从语料库中分析、归纳以及总结得到模型统计信息，然后依据待标注词汇串所处的文本上下文环境计算得到该词汇是某一词性的概率。该方法相对基于规则的方法更加客观（依据统计信息计算得到），可以有效的避免专家预先定义规则时的主观因素。但是由于本方法中使用的模型信息是基于统计学处理的，因此其计算速度要低于基于规则的词性标注方法。本方法对词性的标注效果主要依赖于训练语料的规模，如果语料充分那么模型的计算结果较好。如Moon等人提出了一种基于HMM模型的词性标注方法，文章依据文本内容和功能词间的不同对词汇进行词性标注，使得词性标注准确率得以提高[31]。袁里驰进一步提出了一种改进的基于HMM模型的词性标注方法，通过结合马尔可夫族模型和句法分析使得分词效果得到明显改善[32]。

(3) 规则和统计相结合的词性标注方法

为进一步提高词性标注的准确度，部分学者将上述两种方法进行有机的结合。

规则和统计相结合的词性标注方法不但可以充分利用现有的规则，而且也可以结合使用统计学的模型的客观性和高覆盖率的优势。两种方法结合使用达到互补的效果，使得词性标注的性能得到进一步提升。本方法的基本思想是首先对大规模的训练语料进行统计和分析，从该语料中获取统计模型此外人工从训练语料中归纳抽取规则，然后同时利用规则和统计模型对文本中的词汇进行词性标注。如姜尚仆等人提出的统计和规则相结合的针对日语进行词性标注的系统。虽然文章中提出的方法主要是针对日语的，但是该方法的思想对中文词性标注的研究同样具有重要的借鉴和参考意义[33]。

表 2-1 词性标注方法比较

| 方法名称 | 效率 | 正确率 | 实用水平 |
|---|---|---|---|
| 基于规则的方法 | 高 | 较低 | 不实用 |
| 基于统计的方法 | 较低 | 较高 | 基本实用 |
| 规则与统计相结合的方法 | 较高 | 高 | 实用 |

### 2.2.3 兼类词

单性词的词性标注相对兼类词的词性标注更加简单，对于单性词只需要依据词典进行简单机械配对，然而针对兼类词词性的处理却相对更加复杂。由兼类词的性质可以知道其在不同语境中可能具有不同的词性语法功能，因此针对兼类词的词性标注问题不能像处理单性词那样简单的匹配处理，应该结合词汇所在的上下文环境进行词性的标注，为待标注词汇选择与当前上下文环境相匹配的词性。同时由于中文兼类词现象普遍，因而兼类词的处理是词性标注重要工作，同时也是科研人员主要的研究内容。虽然兼类词只是词汇中的一小部分，但是在中文文本中兼类词的使用却相当频繁。兼类词主要集中在名词与动词，形容词与名词以及动词与形容词等类型上[34]。如下表 2-2 列举了三种类型的词性兼类问题。

表 2-2 兼类词例句

| 名词和动词 | 形容词和名词 |
|---|---|
| 兼类词：空袭<br>名　词：以色列对加沙发动空袭。<br>动　词：以色列空袭加沙。 | 兼类词：秘密<br>形容词：他秘密潜伏在敌营。<br>动　词：我知道你的秘密。 |

## 2.3 依存句法分析

### 2.3.1 依存句法分析概述

依存句法通过分析语言单位（词汇）各成分间的依赖关系从而揭示语句的句法结构。直观的说依存分析的主要目的是识别语句中的“主谓宾”、“定状补”等语法成分，同时分析各成分间的关系。依存句法中动词 为核心词，不受语句中其他成分支配。

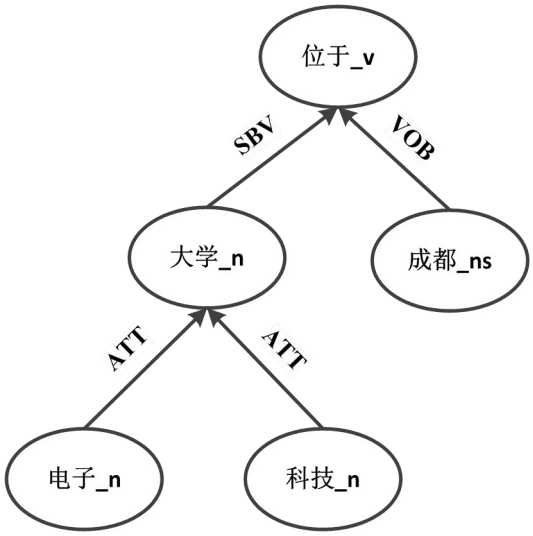


图 2-1 依存分析结果

依存关系表示词与词之间的关系，两个词汇分别称为核心词与修饰词。依存分析的主要目的是对待处理语句进行分析，得到语句中词汇间的依存关系。其中输入的语句可以被表示为 $sen = w_0 w_1 ... w_n$，$w_j$ 表示输入句子的第j个词（$w_j$ 也可以认为是包含第j个词以及该词的词性）。依存树可以形式的地表示为 $d = \{(h,m,r) : 0 \le h \le n, 1 \le m \le n, r \in R\}$，其中 $(h,m,r)$ 表示一个从核心词 $w_h$ 到修饰词 $w_m$ 的依存关系弧，r是 $w_h$ 和 $w_m$ 之间的关系类型，R表示所有关系类型集合。如上图 2-1 是Zpar依存分词得出的依存树，图中的边表示词语之间的依存关系，结点中每个词汇后面的标签是其词性标注的结果。

### 2.3.2 依存句法分析研究现状

20 世纪 70 年代Robinson等人提出了关于依存关系的四条公理。随后，冯志伟等人在Robinson提出的四条公里的基础上进一步提出了第五条公理[35]，进而依存句

法分析得到更快速的发展。一般来说，现阶段依存句法分析主要有基于图模型方法和基于转移的方法。由于第一种依存分析方法是在所有的结果集合中寻找最优解，导致该方法耗费的时间较多、效率降低。由于本方法是寻找全局最优解所以其不管在短距离依存关系还是长距离依存关系中性能都较好。与第一种方依存分析方法相比，第二种方法的解码的速度相对更快，但是其在长距离依存关系上效果不明显[36]。

(1) 基于图的依存句法分析方法

本方法主要是将依存句法树的分析和构建任务，转化为从句子对应的完全有向图中搜索出概率最高的依存树的任务[39]。为了保证全局搜索算法的工作效率，基于图模型的依存分析方法需要如下相关假设：语句对应的依存句法树中，只有子树中的依存关系弧才相互联系、互相影响，而其他的依存关系弧彼此独立。基于图模型的依存句法分析方法主要关注的问题是如何在模型中添加更多的特征时不会导致系统的复杂度增加。本方法的主要改进思路是在增加更多更复杂的特征的同时不断减弱依存子结构的独立性假设。目前为止，经典的基于图的依存分析方法主要有如下几种：2005 年McDonald等人[37] 首次提出了一种基于图的一阶子结构，本方法是基于所有依存关系弧之间是相互独立的假设。随后，McDonald等人[38]又提出了基于图的二阶依存子结构，本方法弱化了独立性假，允许在根节点同一侧的相邻兄弟节点不相互独立，依存分析的性能得到改善。2010 年Koo等人[39] 又提出了基于图的三阶依存子结构，加入祖孙兄弟子树结构特征，进一步提高了依存分析的准确率。

(2) 基于转移的依存句法分析方法

本方法对待处理语句按照一定的顺序处理通常是从左到右，该过程可以被视为一个动作序列，动作序列不同产生的依存分析树的结构也不同。因此可以将依存句法分析的问题转化为搜寻最优动作序列的问题。该方法相对于基于图的方法，可以更加方便简单的加入新的特征。由于本方法的解码时间是线性的，因此本方法不会随着新特征的加入而使系统的效率大大降低。目前本方法中最主流的是基于栈的转移系统，如arc-Standard算法。

为了寻求中文自然语言处理更优的结果，大批国内外研究者提出和实现了许多自然语言处理工具，其中被广泛使用的系统主要有：中科院的 ICTCLAS 系统、复旦大学的Fudan NLP自然语言处理系统[40]、哈尔滨工业大学社会计算与信息检索研究中心发布的语言技术平台[41](LTP,Language Technology Platform)、斯坦福大学的Stanford-Parser以及Zpar[42]等。

## 2.4 机器学习相关算法

### 2.4.1 聚类

#### 2.4.1.1 聚类概述

聚类是指按照事物的相关属性，将事物聚集为一个一个的类别，使得各类别内部所有对象间相似性较高，而不同类别的对象间的彼此相似程度较低。聚类结果质量主要和聚类算法中使用的距离和相似性度量方法相关。

为了度量聚类对象间的相似程度，一般采用距离和相似度进行表示。通常对象间距离越接近，那么对象就越相似，则他们是同类的可能性就更大。常用的相似度和距离定义有:余弦相似度，相关系数，欧几里得距离，曼哈顿距离和切比雪夫距离等。下面用两个m维向量 $X=(x_1,x_2,.....,x_m)$ 和 $Y=(y_1,y_2,.....,y_m)$ 表示上述距离和相似度计算公式，$S_{ij}$ 表示两个向量间的相似度，$d_{ij}$ 表示两个向量之间的距离。余弦相似度：

$$S_{ij}(X,Y)=cos(\theta)=\frac{\sum_{k=1}^{m}x_k y_k}{\sqrt{\sum_{k=1}^{m}x_k^2}\sqrt{\sum_{k=1}^{m}y_k^2}} \tag{2-1}$$

相关系数的计算公式:

$$S_{ij}(X,Y)=\frac{\sum_{k=1}^{m}(x_k-\bar{X})(y_k-\bar{Y})}{\sqrt{\sum_{k=1}^{m}(x_k-\bar{X})^2}\sqrt{\sum_{k=1}^{m}(y_k-\bar{Y})^2}} \tag{2-2}$$

曼哈顿距离:

$$d_{ij}(X,Y)=\sum_{k=1}^{m}|x_k-y_k| \tag{2-3}$$

欧几里得距离:

$$d_{ij}(X,Y)=\sqrt{\sum_{k=1}^{m}|x_k-y_k|^2} \tag{2-4}$$

切比雪夫距离：

$$d_{ij}(X,Y)=\max_{1\le k\le m}|x_k-y_k| \tag{2-5}$$

明科夫斯基距离

$$d_{ij}(X,Y)=\left(\sum_{k=1}^{m}|x_k-y_k|^q\right)^{1/q},q>0 \tag{2-6}$$

当明科夫斯基距离中q为 0，1 以及无穷大则分别对应曼哈顿距离，欧几里得距离以及切比雪夫距离。

#### 2.4.1.2 聚类研究现状

由于现实中数据的复杂性，所以现阶段没有一种普遍的聚类算法可以用来处理各种数据集。对于聚类算法的选择一般需要依据待处理数据的特点、聚类目的以及实际的用处。下文将对主要的聚类方法进行分别介绍，分类如下图 2-2 所示。

(1) 基于划分的方法

基于划分的聚类方法是将具有n个对象的数据集合划分为k个子集，其中k小于等于n，每个划分的子集代表一个类别即聚类结果中的一个簇。基于划分的聚类方法中最典型是K-means算法[43]。

(2) 层次聚类方法

层次聚类是指对数据进行逐层的地进行处理，可以自底向上地把多个小的类别合并起来（凝聚层次聚类），同时也可以自顶向下地将大的类别分割为多个小的类别（分裂层次聚类）[44]。凝聚层次聚类方法是指先将数据集合中每个对象看成是独立的一类，然后将这些小的类别进行合并，如此不断迭代直到满足终止条件，算法结束，该类的代表算法有AGNES算法。分裂层次聚类方法首先将所有数据对象视作一个大的类别，然后将这个大的类别不断的进行划分，如此不断的迭代直到满足终止条件算法结束，该类的代表算法有DIANA算法。

(3) 基于密度的方法

基于密度的聚类方法是通过数据对象的密度来对数据进行类别划分的。该方法将数据组成的一个个的类别对象可以看做是空间中被分隔开的不同区域。本方法中，如果数据集中的邻近点没有超过预先定义的阈值就需要继续进行聚类，这样不断继续下去将会形成众多的类别[45]。GDBSCAN、DENCLUE是典型的基于密度的聚类方法。GDBSCAN方法主要是通过检测每个数据对象邻居的个数是否超过预先定义的阈值，如果超过预先定义的阈值时则认为这个对象周围的数据已经足够。与GDBSCAN方法不同DENCLUE方法主要是通过影响函数取定空间的密度。DENCLUE方法使用树型存储以及网格单元，其处理速度快，能够处理高纬度的数据，同时其对参数异常敏感。

(4) 基于网格的方法

该类方法是指使用网格技术将数据对象划分为一个个网格结构，然后在网格数据进行聚类。该算法的优点是聚类的速度与数据对象的数量无关，仅仅和网格的数目有关。该方法聚类结果的质量主要取决于划分的网格结构的底层粒度，其粒度越细，处理的代价就越大。此外，在划分上层单元时，统计信息网格没有考虑底层以及其相邻单元间的关系。尽管这种方法处理的速度较快，但是其聚类结果的质量和精确性将会受到影响[46]。典型的基于网格的聚类方法有WaveCluster方法和CUQUE方法。

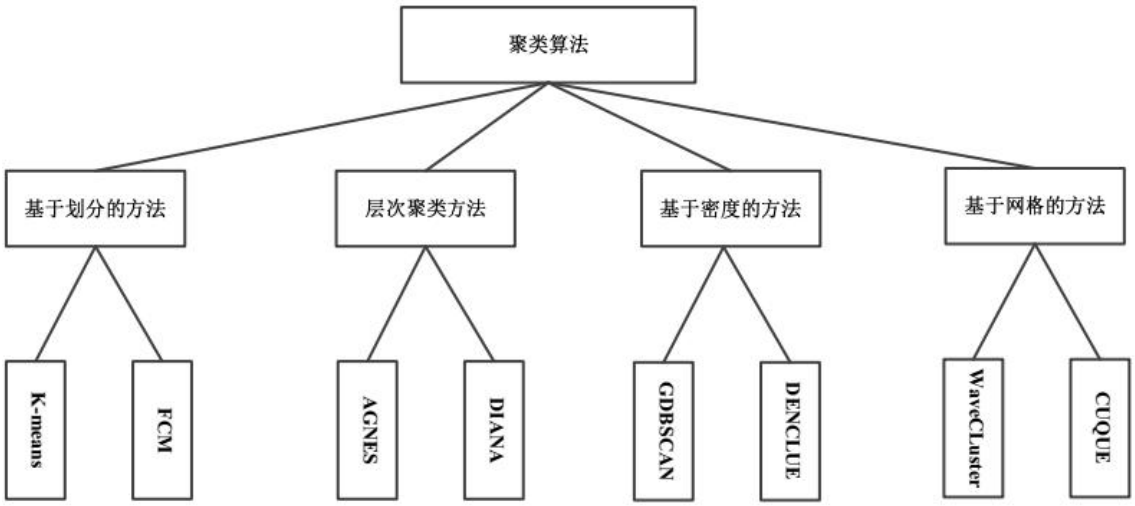


图 2-2 聚类算法分类

### 2.4.1.3 K-means 算法

K-means是一种基于划分的聚类算法，其采用距离来度量数据之间的相似性。该算法的核心思想就是如果两个数据对象的距离越接近，那么这个两个数据对象的相似性就越大。K-means是麦克奎因于 1967 年提出，由于该算法的简单性，因此其被广泛应用于学术界和页界。

K-means具体过程如下：第一步，随机预先指定k个点作为每个类别的初始中心；第二步，计算各个样本数据到指定的类别的初始中心的距离，其中把样本数据划分为离它最近的那个类别中心所在的类，这样就产生了初始的k个类别分布;第三步，对调整后的新的数据类别分布重新计算新的聚类中心，然后重复执行第二步的划分过程，这样经过若干次的迭代后，如果类别中心不再改变，说明函数收敛，聚类结束，否则将继续执行第二步和第三步直至收敛。K-means在每次迭代计算的过程中都需要验证所有数据对象的类别是否正确，如果不正确就需要进行

调整。当全部数据样本重新调整之后，再修改类别的中心，之后进行下一次的迭代。K-means算法的伪代码如下：

**K-means**

输入：n 个数据对象 xi 集合 X

输出：k 个聚类中心 Cj 以及 k 个类别数据集合 Dj

算法：

Begin

flag=true

initial k prototype Cj， $j \in [1,k]$

while(flag)

for $i \leftarrow 1$ to n do

$D(x_i, c_j) \leftarrow |x_i - c_j|$

if $D(x_i, c_j) = \min\{D(x_i, c_j)\}$

then insert xi into Dj

if Cj changed

$$c_j \leftarrow \frac{1}{N_j}\sum_{p=1}^{N_j} x_{jp}, j \in 1,2,.....,k$$

else

flag=false

END

## 2.4.2 Logistic 回归

Logistic 算法是成熟且被广泛使用的机器学习算法。其可以计算得到样本数据所属各个类别的概率进而对样本所属的类别进行预测。下文将对 Logistic 算法进行解释以及推导。

为了方便解释，假设存在 $n$ 个训练数据 $X = \{x_1, x_2, ...x_n\}$，且其对应的类型标签为 $Y = \{y_1, y_2, ..., y_n\}$，其中每个样本 $x_i$ 的维度为 $d$，其形式为 $x = [x^{(1)}, x^{(2)}, ...x^{(d)}]^T$，$y_i \in \{1, 2, ..., c\}$。Logistic 算法预测函数如公式 2-7，向量 $\theta$ 是需要学习的参数。

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}} \tag{2-7}$$

其中

$$g(z) = \frac{1}{1+e^{-z}} \tag{2-8}$$

是 Sigmoid 函数。从该函数的性质中可以得到 $g(z)$ 的取值范围是[0,1]，因此 $h_\theta(x)$ 的取值范围也是[0,1]。

为方便实际使用，在 $\theta^T x$ 的后面增添常数 $\theta_0$，为保持参数 $\theta$ 的结构一致，需要使得 $x^{(0)}=1$，则训练样本可以改为如下形式 $x=[x^{(0)},x^{(1)},...,x^{(d)}]^T$，即 $\theta^T x=\sum_{j=0}^{d}\theta_j x^{(j)}$。

对于二分类问题，如果给定的样本 $x$，其类别标签 $y=1$ 的可能性为：

$$P(y=1\mid x,\theta)=h_\theta(x)=\frac{e^{\theta^T x}}{1+e^{\theta^T x}} \tag{2-9}$$

则样本 $y=0$ 的可能性为：

$$P(y=0\mid x,\theta)=1-h_\theta(x)=\frac{1}{1+e^{\theta^T x}} \tag{2-10}$$

则对于给定的样本 $x$，换种表达形式其分类可表示为：

$$P(y\mid x,\theta)=h_\theta(x)^y(1-h_\theta(x))^{1-y} \tag{2-11}$$

算法的学习过程就是通过计算得到 $\theta$，公式 2-11 表示后验概率 $P(y\mid x,\theta)$ 的计算，该结果对于预测分类有很大的影响。具体地，如果 $P(y\mid x,\theta)$ 的数值比某个预设的阈值大，那么认为该样本分类结果是 1，不然认为样本的类别是 0。

为了计算参数 $\theta$，本算法首先需要定义损失函数，然后通过最小化损失函数或者极大似然估计求解出 $\theta$，下文将使用极大似然的方法求解出参数。假设上面定义的样本 $X$ 和其所述类别 $Y$ 之间相互独立，那么每一样本边际分布的乘积即为该样本的联合分布，关于 $\theta$ 的极大似然估计函数可以用公式 2-12 表示：

$$L(\theta)=\prod_{i=1}^{n}P(y_i\mid x_i,\theta)=\prod_{i=1}^{n}h_\theta(x_i)^{y_i}(1-h_\theta(x_i))^{1-y_i} \tag{2-12}$$

然后对极大似然函数取对数，通过最大化对数似然函数计算 $\theta$ 的值，$\theta$ 的对数似然函数为：

$$\begin{aligned} l(\theta)&=\log L(\theta)=\log\Big(\prod_{i=1}^{n}h_\theta(x_i)^{y_i}(1-h_\theta(x_i))^{1-y_i}\Big)\\ &=\sum_{i=1}^{n}\big(y_i\log h_\theta(x_i)+(1-y_i)\log(1-h_\theta(x_i))\big)\end{aligned} \tag{2-13}$$

然后采用梯度下降求解 $\theta$，具体是对 $\theta$ 的各个分量求偏导，即：

$$\begin{aligned}\frac{\partial}{\partial\theta_j}l(\theta)&=\frac{\partial}{\partial\theta_j}\sum_{i=1}^{n}(\mathrm{y}_i\log h_\theta(x_i)+(1-\mathrm{y}_i)\log(1-h_\theta(x_i)))\\&=\sum_{i=1}^{n}(\mathrm{y}_i\frac{1}{h_\theta(x_i)}-(1-\mathrm{y}_i)\frac{1}{1-h_\theta(x_i)})\frac{\partial}{\partial\theta_j}h_\theta(x_i)\\&=\sum_{i=1}^{n}(\frac{\mathrm{y}_i}{h_\theta(x_i)}-\frac{(1-\mathrm{y}_i)}{1-h_\theta(x_i)})h_\theta(x_i)(1-h_\theta(x_i))\frac{\partial}{\partial\theta_j}\theta^Tx_i\\&=\sum_{i=1}^{n}(\mathrm{y}_i(1-h_\theta(x_i))-(1-\mathrm{y}_i)h_\theta(x_i))x_i^{(j)}\\&=\sum_{i=1}^{n}(\mathrm{y}_i-h_\theta(x_i))x_i^{(j)}\end{aligned}\tag{2-14}$$

在公式 2-14 中，$x_i^{(j)}$ 表示第 $i$ 个样本的第 $j$ 个特征。在公式 2-15 中，$\alpha$ 表示学习速度，这一数值随着实际需求的不同而做出改变，从而选出一个合适的数值。以此为基础，针对 $\theta$ 的各分量 $\theta_j$ 为：

$$\theta_j:=\theta_j+\alpha\sum_{i=1}^{n}(\mathrm{y}_i-h_\theta(x_i))x_i^{(j)}\tag{2-15}$$

多分类问题即 $y_i\in\{1,2,...,\mathrm{c}\}$ 可以由二分类器利用一对多的策略变换拓展而来的。具体的过程是在 $c$ 个类别中随机选择一个类作为一个类别，然后将其他剩余的 $c-1$ 个类看成一个大类，像这样构造多个二分类器。基于一对的方法共构建了 c 个分类器，最后使用投票的方式进行对结果进行统计进而实现多分类。对于多分类，可以表示为：

$$\begin{aligned}&P(y=1\mid x)=\frac{e^{\theta_1^Tx}}{1+\sum_{k=1}^{c-1}e^{\theta_k^Tx}}\\&\cdots\\&P(y=c-1\mid x)=\frac{e^{\theta_{c-1}^Tx}}{1+\sum_{k=1}^{c-1}e^{\theta_k^Tx}}\\&P(y=c\mid x)=\frac{1}{1+\sum_{k=1}^{c-1}e^{\theta_k^Tx}}\end{aligned}\tag{2-16}$$

定义损失函数为：

$$J(\theta)=-\frac{1}{n}\left(\sum_{i=1}^{n}\sum_{k=1}^{c}I\{\mathrm{y}_i=\mathrm{k}\}\log\frac{e^{\theta_k^Tx}}{\sum_{l=1}^{c}e^{\theta_l^Tx}}\right)\tag{2-17}$$

其中：$I\{\mathrm{y}_i=\mathrm{k}\}$ 表示指示器函数，若 $\mathrm{y}_i=\mathrm{k}$，则 $I\{\mathrm{y}_i=\mathrm{k}\}=1$；否则 $I\{\mathrm{y}_i=\mathrm{k}\}=0$。更多有关 Logistic 算法的详细信息请参考[47]。

## 2.5 本章小结

本章介绍了相关技术背景及算法，主要介绍了自然语言预处理的分词、词性标注、依存分析，同时介绍了机器学习相关算法包括包括聚类算法 K-means 和 Logistic。本文介绍的相关技术和算法对本课题的研究有很重要的影响，并且也被广泛使用，对本文提出新算法有很大的帮助。

# 第三章 中文开放式实体关系抽取算法研究

## 3.1 关系抽取算法流程

开放式关系抽取不像传统的关系抽取方法需要判断句子中的实体对是否存在某种预先定义的关系类型，而是直接从文本中抽取表示关系的关系指示词从而得到实体关系元组（实体 1，关系指示词，实体 2），因此本方法抽取的关系更具有普遍性，更易理解。开放式关系抽取不用预先定义关系类型，可以抽取得到丰富的实体关系，这对于理解和应用海量 Web 数据具有重要的意义。基于此，由于本文是从网络海量文本数据中抽取关系元组，所以文本的获取和处理是我们面临的第一个问题。然后，通过对获取的文本数据抽取丰富的候选实体关系元组。最后，为确保抽取的关系元组的可靠性，本文使用机器学习的方法对抽取得到的候选关系元组进行置信度评估，并过滤低质量的关系元组。

如图 3-1 所示，本文提出的中文开放式实体关系抽取算法 DPM（Disambiguated Pattern Matching Model for Chinese Relation Extraction）的基本框架主要包括如下步骤：关系模式抽取（见 3.2 节）、候选关系元组抽取（见 3.3 节）、关系元组置信度评估和过滤（见 3.4 节 ）。后文将详细讨论这三个主要关系抽取步骤。

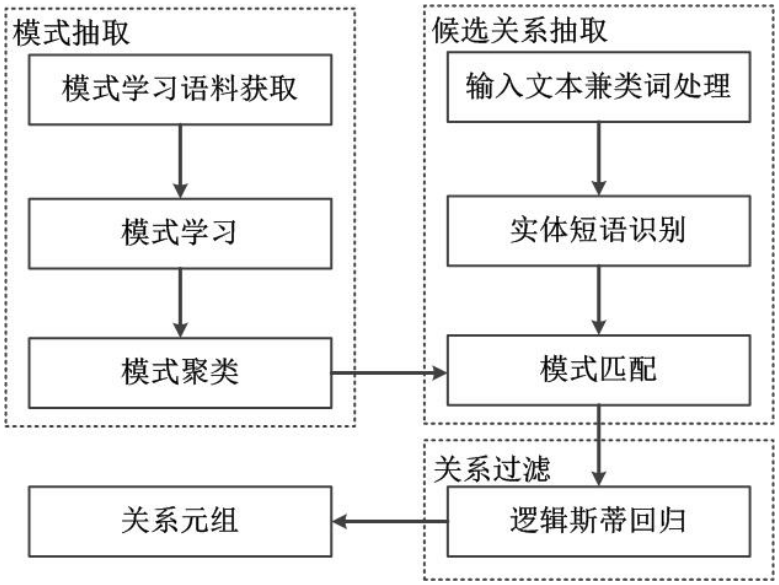


图 3-1 中文开放式实体关系抽取系统结构图

## 3.2 关系模式抽取

本模块是利用大规模高质量关系元组和对应语句的训练语料，抽取得到实体

和关系词之间的依存路径与词性的关系模式。关系模式获取的具体步骤如图 3-2 所示，包括如下三个步骤：首先获取训练语料，然后对训练语料进行预处理并抽取依存路径以及词性关系模式，最后对抽取的关系模式进行聚类。

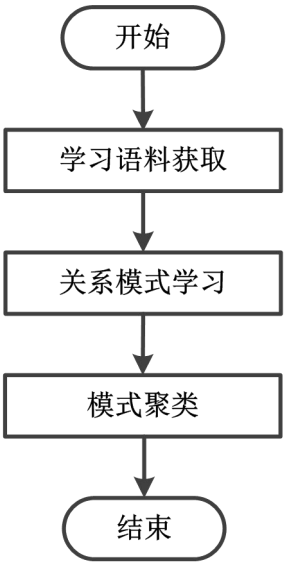


图 3-2 模式学习流程图

### 3.2.1 训练语料的获取

本节所指的训练语料是指由关系元组以及包含该关系元组的语句组成的语料，即一条语料包括两个部分：关系元组和相应的原始语句。例如从如下语句“奥巴马当选为总统”中可以抽取出如下的实体关系元组(奥巴马，当选，总统)，则该关系元组和该条语句共同构成一条训练语料。本文首先通过百科信息框、关系知识库以及利用已有关系抽取算法获取高质量的关系元组，然后获取训练语料。

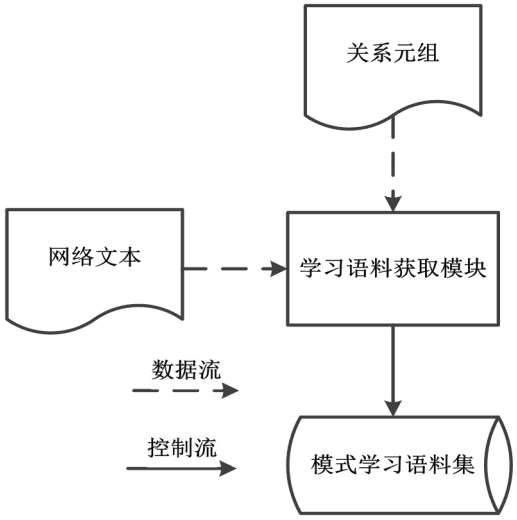


图 3-3 训练语料获取结构图

由于网络百科页面中的信息框中包含大量的属性关系元组，这为自动构建用于学习关系模式的语料提供了重要的基础。因此，构建训练语料的一种重要的方法是利用百科页面信息框中的属性关系元组，然后通过网络爬虫获得包含关系元组中实体和关系词的对应语句。具体的流程如上图 3-3。

百科页面信息框中的关系元组的获取具体见 5.5 节。如“奥巴马”百度百科信息框，经过爬虫处理然后解析得到多个关系元组如(奥巴马，中文名，贝拉克·侯赛因·奥巴马)，(奥巴马，别名，欧巴马)，(奥巴马，妻子，米歇尔·拉沃恩·奥巴马)等。为获取包含关系元组所对应的语句，本文通过使用搜索引擎寻找包含关系元组的语句。在获取包含关系元组的候选网页后，然后爬取并解析获取其中频率最高的语句，如果同时存在多个频率相同的语句，则选择其中较短的语句。如下图 3-4 所示为搜索(奥巴马，妻子，米歇尔·拉沃恩·奥巴马)关系元组所示的网页页面，经过抽取，此关系元组得到的对应语句为图中圈出部分。通过这种方式即获得了一条用于抽取关系模式的训练语料。


图 3-4 （奥巴马，妻子，米歇尔·拉沃恩·奥巴马）搜索结果

此外对于高质量的关系元组也可以通过其他方式获取，如直接利用关系知识库中已有的高质量的关系元组或者采用已有的中文关系抽取系统例如 ZORE 系统，

从大规模开放语料中抽取实体关系元组，并选择其中置信度较高的关系元组。

### 3.2.2 关系模式学习

本模块主要是利用大规模高质量的训练语料初步抽取实体和关系词之间的依存路径和词性模式。首先利用自然语言处理工具，如Zpar对上一步骤中获取的高质量训练语料中的语句进行分词、词性标注以及依存分析。然后抽取关系元组中实体和关系词在原语句中的词性，以及实体和关系词之间的依存路径，作为关系模式。此外在抽取关系模式的同时记录了模式的频率。本文中抽取得到的模式的形式为：实体和关系词间的依存路径以及路径中词汇的词性。例如，通过例句“巴育当选为总理”和其对应的关系元组(巴育，当选，总理)可以学习得到关系模式“SBV(nr)-ROOT(v)-VOB(n)”。首先通过Zpar对例句进行依存分析得到 3-5 依存分析树，其中“巴育”和“总理”是关系元组的实体，“当选”是关系元组的关系词；然后结合例句的依存结构和关系元组中的成分得到关系模式“SBV(nr)-ROOT(v)-VOB(n)”，其中“ROOT(v)”表示关系，“SBV(nr)”和“VOB(n)”表示关系元组中的实体。本文通过初步处理，抽取得到 3769 个关系模式。

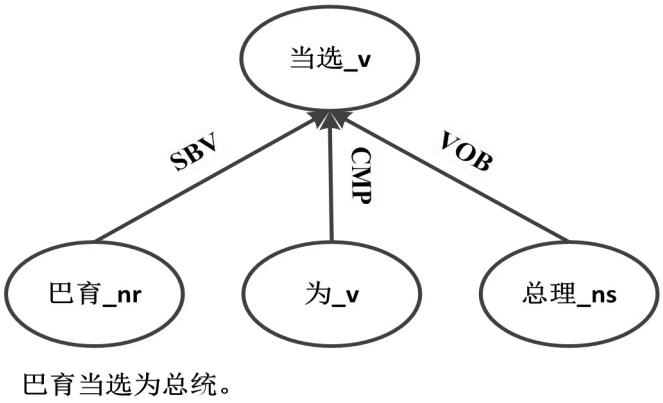


图 3-5 依存分析结构图

### 3.2.3 模式聚类

考虑到通过大规模学习得到的依存路径和词性的关系模式可能存在分歧和矛盾(例如某些具有细微差别的模式其实具有相同的语法含义)，本文提出进一步对学习得到的模式进行聚类，以提高模式对复杂自然语言环境的适用性和覆盖率。本文采用K-means算法对初步抽取的 3769 个模式进行聚类。

在得到初步抽取的关系模式集合后，为了将关系模式用于聚类，需要将关系模式中依存分析标签和词性标注标签转化为数值，构建为表示关系模式的特征向量。在对关系模式进行数值化时，考虑到关系模式中不同的依存关系和词性对模

式的判别所作出的贡献不同，需要为模式中每个依存关系和词性一个不同的数字表示其对应的编码。由于依存关系和词性的个数是有限的，所以为关系模式进行编码是可行的。依存分析标签和词性标注标签对应的编码如 5.2 节中表 5-1 和表 5-2 所示。如“SBV”的编码为 02，“nr”的编码为 24，关系模式“SBV(nr)-ROOT(v)-VOB (n)”的编码为 02(24)-01(15)-03(23)。

得到关系模式的编码后，为进一步进行聚类操作，首先对关系模式的编码进行预处理分别得到依存模式向量和词性标注向量，然后将依存模式向量相同的关系模式作为一个大的类别，最后再依据词性标注向量对其进行聚类。例如，在初步抽取的模式库中存在一个与上文中的关系模式“SBV(nr)-ROOT(v)-VOB(n)”相似的关系模式“SBV(n)-ROOT(v)-VOB(n)”。该关系模式对应的模式编码为 02(23)-01(15)-03(23)。对其编码进行切分得到依存模式的向量(02,01,03)和词性标注的向量(23,15,23)。同时对上文中的模式编码 02(24)-01(15)-03(23)进行切分得到依存模式向量(02,01,03)和词性标注向量(24,15,23)。这两个关系模式的依存关系向量相同，然后对他们的词性标注向量进行聚类。如下算法描述了本文关系模式的聚类过程。

**模式聚类**

输入：ModelSet 初步抽取的所有的关系模式编码集合

输出：Result[i]最终输出结果，k 个聚类的数据集合

```
Begin
        //本行是获取关系模式编码集合中所有不同的依存关系向量集合
        DenVecSet ← getDenVecSet(ModelSet)
        for  model  in  ModelSet
            for  denVect  in  DenVecSet
                If  denVect=getDenVec(model)
                    //getPosVec(model)是获取某个模式的依存关系向量
                    PosVec ← getPosVec(model)
                    //获取词性标注向量并加入多个词性标注集合中
                    DataSet[i].add(PosVec)
        for  i ← to k do
            //分别对多个词性标注集合数据进行聚类
            Result[i] ← K-means(DataSet[i])
END
```

K-means聚类代价和聚类的类簇个数间的关系如图 3-6 所示，横坐标表示聚类

的类簇个数，纵坐标表示对应类簇个数时总代价（本文使用欧式距离）。图中可以看出，对于在特定的依存模式下，其中所有的词性标注模式都被聚在一个类别中。这主要是由于实体关系元组中实体都是名词性词汇，同时在某个固定的依存模式下其关系词的词性也是固定的（动词或者名词中的一种）。聚类之后，通过观察同一类别中大量关系模式，可以发现一些实际含义相同但形式不同的模式，如“SBV(n)-ROOT(v)-VOB(n)”，“SBV(nr)-ROOT(v)-VOB(n)”，“

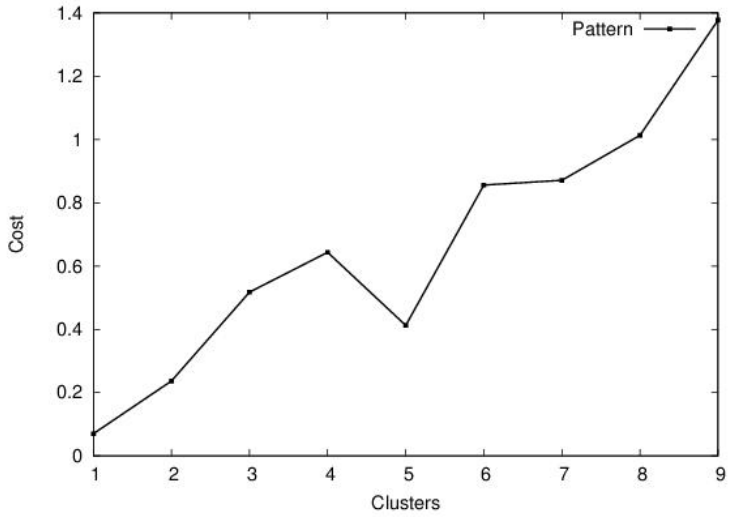


图 3-6　K-means 分析

SBV(nt)-ROOT(v)-VOB(n)”等。通过对聚类结果的分析，可以看到当依存模式相同的时候，“j”、“i”、“nr”、“ns”、“nt”、“nz”、“r”等词性的模式被聚合在一起，这些词性具有和名词“n”相同的性质。这进一步验证了关系元组中的实体是名词性成分，因此这种聚类结果是符合认知的。经过聚合处理，本文从 3769 个关系模式中得到 2431 个关系模式，其中有 621 个高频率的关系模式。如下表 3-1 中展示了 6 个频率最高的的关系模式，而且这 6 个高频率的关系模式是符合认知的。其中，第一列是关系元组，第二列是关系模式。

表3-1 关系模式

| 关系元组 | 关系模式 |
|---|---|
| (IC(v),SBV(n),VOB(n)) | IC(v):SBV(n):VOB(n) |
| (ROOT(v),SBV(n),VOB(n)) | ROOT(v):SBV(n):VOB(n |
| (IC(v),SBV(n),POB(n)) | IC(v):SBV(n):ADV(p),POB(n) |
| (ROOT(v),SBV(n),POB(n)) | ROOT(v):SBV(n):ADV(p),POB(n) |
| (APP(n),ATT(n),ROOT(n)) | APP(n):ATT(n):ROOT(n) |
| (IC(v),SBV(n),ADV(t)) | IC(v):ADV(t):SBV(n) |

## 3.3 候选关系抽取

本节主要是利用抽取得到的高质量的关系模式对待抽取文本进行模式匹配并抽取候选实体关系元组。首先对待抽取语句使用相关工具进行分词、词性标注以及依存分析；然后为提高分词以及词性标注的准确性，本文进一步对兼类词进行处理；最后把待抽取语句和学习得到的关系模式进行匹配抽取关系元组。如图所示，本模块主要包括如下步骤：兼类词处理、候选实体识别、模式匹配与关系抽取、关系扩展。

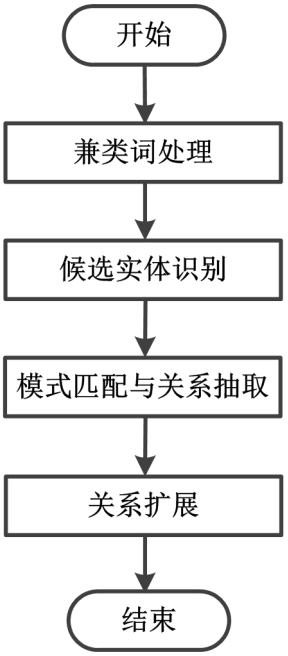


图 3-7 候选关系抽取流程图

### 3.3.1 兼类词处理

本模块首先使用 Zpar 自然语言处理工具对待抽取文本进行分词、词性标注和依存分析，然后再处理兼类词。兼类词是指某个词在不同的上下文环境中拥有两种或多种语法功能，即该词汇在不同上下文环境中拥有不同词性。与英文相比，英文中词根相同但词性不同的单词通常具有不同的词形，而中文环境下，词性兼类现象则较为常见。由于中文词性兼类现象的普遍性和复杂性，因此兼类词的处理能够有效提高文本预处理阶段词性标注准确度，并且可以广泛应用于各种中文信息处理系统中。

本文提出的兼类词处理方法具体包括如下步骤：首先从海量文本中选择出大量包含兼类词的语句并通过分析得到具有较高准确率和覆盖率的兼类词语义角色统计规则；然后对待处理文本进行自然语言预处理，并基于词法、句法和语境规

则库对分词结果中的兼类词词性进行标注；最后使用兼类词依存语义角色统计规则进一步准确识别兼类词在不同上下文环境中的词性。

(1) 兼类词语义角色规则获取

首先，利用自然语言处理工具对大规模文本数据进行预处理。通过比较现有自然语言处理工具并选择其中准确率和效率相对较高的工具 Zpar 对输入文本中的语句进行分词、词性标注以及依存分析等一系列的自然语言预处理。然后，通过人工统计语料中兼类词依存分析结果，得到如下中文兼类词语义角色统计规则。

兼类词语义角色规则 1：

若一个词语的词性标签标记为 v，且其依存分析语义角色为“ATT”，则该词的词性应调整为名词。

兼类词语义角色规则 2：

若一个词语词性标签为非名词性，且其依存标签为“VOB”，“POB”或者“IOB”，且没有依存标签为“VOB”，“POB”或者“IOB”的节点与之直接相连或者通过介词相连，则该词的词性应调整为名词。

(2) 词法、句法和语境规则库兼类词词性处理

在对待处理文本数据进行自然语言预处理后，使用统计得到的词法、句法、语境规则库初步识别待抽取语句中兼类词词性。本文主要使用并列类推、同语境类推、“有”的宾语是名词等规则识别兼类词词性。并列类推规则是指并列词汇的词性相同，因此可以依据其中一个词汇的词性得出另外一个词汇的词性，如“人民的要求和愿望”，依据并列规则得到兼类词“要求”在此上下文中的词性与“愿望”的词性相同，因此兼类词“要求”的词性是名词；同语境类推规则是指对于相同语境下的词汇其词性相同，因此可以依据其中一个词汇的词性得到另外一个词汇的词性，如“好材料、好设计”，依据该规则兼类词“设计”的词性和“材料”的词性相同，因此兼类词“设计”的词性是名词；“有”的宾语是名词，如“有希望”则其中的词汇“希望”的词性为名词。

(3) 依存语义角色兼类词词性处理

在词法、句法和语境规则库初步识别语句中兼类词词性的基础上，进一步使用预先得到的兼类词依存语义角色规则识别该语句中其他未识别的兼类词的词性。具体的是将语句中的所有词汇的依存分析结果、词性分别与兼类词语义角色统计规则进行匹配，若符合其中任意一条规则，则对兼类词词性做相应的调整。如下图 3-8 为“奥巴马总统与中国驻美国大使进行会谈”的依存分析结果，其中“驻”的词性标注结果是“v”且其依存分析结果为“ATT”，则其符合兼类词依存分析语义角色统计规则 1，所以应将其调整为“n”。如下图 3-9 中“选举”的词性标注

结果为“v”且其依存分析结果为“VOB”，且没有依存标签为“VOB”，“POB”或者“IOB”的节点与之直接相连或者通过介词相连，则其符合兼类词依存分析语义角色规则 2，所以应将其调整为“n”。

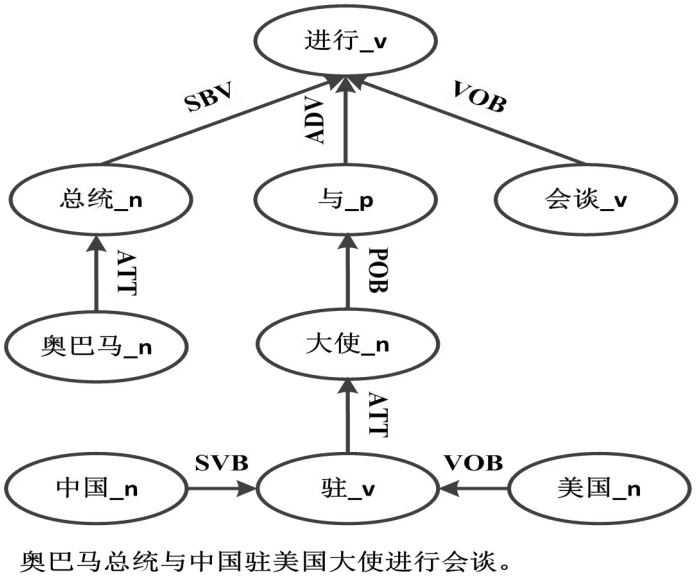


图 3-8 依存分析树，兼类词规则 1

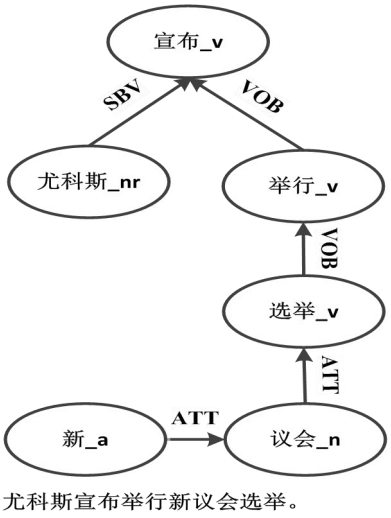


图 3-9 依存分析树，兼类词规则 2

本节通过统计得到高覆盖率和高准确率的兼类词依存分析语义角色规则，并基于此结合词法、句法、语境规则提高了中文兼类词词性识别准确度，因此词性标注的准确度也得到进一步提高，为下一步关系抽取提供了更加良好的自然语言处理基础。

### 3.3.2 实体短语识别

实体对是关系元组中重要的组成部分，因此实体识别的正确与否对关系抽取算法有着重要的影响。在实验中本文使用Zpar进行实体识别，但是发现在对复杂无规律的网络文本进行实体识别的时候，存在很多实体(主要是机构名，人名)难以识别出来或者识别错误。这是由于Zpar中集成的命名实体识别模型使用的训练语料中实体的数量是非常有限的且领域单一的，不会随着的时间而不断的增加，导致其在处理网络文本时机构名实体和人名实体识别的召回率降低。为了提高实体识别的召回率，本文通过从百度百科网页中抽取人名和机构名等实体以及使用开放式实体抽取方法从垂直网站中抽取实体加入到实体识别的字典中。我们通过百度百科页面和垂直网站构建了一个实体字典，有效的提高了实体识别的准确性。

从百度百科获取实体列表主要是基于百度百科的开放分类，然后基于此爬取网页源码，并从中解析出实体加入实体列表中。如下图 3-10 为百度百科的开放分类，图中机构开放分类共 19 个大类，其中包括企业、博物馆、公司等。本文从中大约获取 158000 个机构类的URL，然后设计构建了多线程网络爬虫，其任务是要抓取 158000 个URL中机构网页的源文件并从中解析出机构名并加入实体字典中。


图3-10 百度开放分类

现如今网络中存在各种领域的垂直网站，而且就某一垂直网站而言其对某一领域实体的覆盖率是相当高的，包括了某一特定领域中大部分实体，这对于我们构建实体字典具有重要意义。从这种垂直网站中抽取大量实体，可以认为是面向开放域实体抽取的任务。面向开放域实体抽取是指在给出特定语义类的若干个实

体然后在数据集中找出该语义类包含的其他实体。从抽取的方式上看，传统的实体识别技术更关注的是从文本中识别出实体字符串所在的位置和其所属的类别，而开放式实体抽取技术主要关注从冗余且不规范的 Web 数据中抽取出与预先指定的种子实体具有相同语义类的实体。换句话说二者的区别是：传统的实体抽取算法目标是抽取出文本中出现的所有的实体，而面向开放域的实体抽取是只抽取某个语义类的实体列表(也就是符合某个模式的实体)。但是由于网络中存在海量的数据所以这种方法获取的实体数量巨大。将这些抽取得到的大量的实体加入到 Zpar 的字典中可以进一步提高其实体识别的召回率。如下图 3-11 所示，比如给出奥迪，宝马，奔驰这三个实体可以在垂直网站中找出和其相似的实体，如本田、别克、大众等等。


图 3-11 垂直网站实体抽取

通过以上两种方式大大提高了实体识别的召回率。为进一步识别待抽取语句中候选实体短语，本文通过定义一系列规则得到实体短语。具体的方法是将所有词性标记为“t”、“m”、“q”、“s”、“r”、“u”、“a”以及名词类词性标记(“n”、“nr”、“nz”、“ns”、“nt”)其中之一，且其在句子的依存句法分析树上直接和上文识别的实体相连，则将该词汇和识别的实体进行合并。本文在后面的描述中将单个实体和实体短语统称为实体短语。并且将与关系词距离较近的词看做实体短语核心词。例如，对于句子“奥巴马总统与中国驻美大使进行会谈”，首先得到其依存句法分析树如图 3-8 所示，然后根据该树得到词性标记为名词 “n” 的实体核心词“总统”以及与“总统”相邻的另一个词性标记为名词 “n” 的单词“奥巴马”，合并两个单词得到完整实体短语“奥巴马总统”。

### 3.3.3 模式匹配和关系抽取

抽取候选三元组是基于模式匹配的方法实现的，其是利用3.2节学习得到的关系模式从待处理语句中抽取关系元组。首先对语句的依存句法分析树进行扫描，将依存角色标签“ROOT”、“IC”、“ATT”、“VV”等21个标签中任意一个，且词性是

动词“v”的所有词汇作为该语句的候选关系词。然后从3.2节中学习的关系模式字典中获取候选关系词的所有关系模式，同时获取该候选关系词到句子中所有实体短语的路径的依存标记序列和词性标记序列。如果存在某个序列和某个模式中某一方向上的路径相匹配，则将该序列对应的名词性短语作为候选名词，若存在多个候选名词到关系词间的路径不重复，且覆盖某个模式中所有方向上路径，则抽取这些候选实体短语作为该关系词对应的候选关系元组，再结合关系词得到完整的关系元组。例如，图3-12中句子“普金访问克里米亚”，首先根据标记“ROOT(v)”获得候选关系词“访问”，得到“ROOT(v)”相关的所有模式，然后获取名词性短语“普金”和“克里米亚”到“访问”的序列“SBV(nr)，ROOT(v)”和“VOB(nz)，ROOT(v)”，将其依据聚类的规则处理后得到“SBV(n)，ROOT(v)”和“VOB(n)，ROOT(v)”，分别匹配模式“ROOT(v): SBV(n); VOB(n)”中一个方向上的依存路径，所以将“普金”和“克里米亚”作为候选名词，得到候选关系元组(普金，访问，克里米亚)。

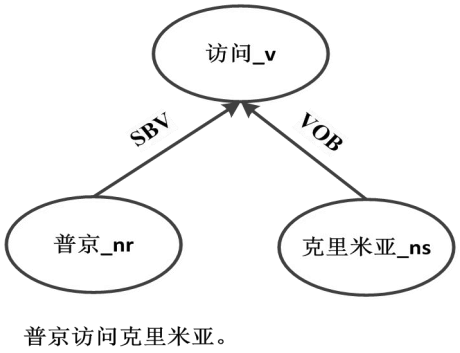


图 3-12 依存分析树

在基于模式匹配实现关系抽取的过程中，会出现对于同一待抽取句子，可能存在多个模式同时适用的问题。针对这种问题，本文根据先验概率进行模式选择。具体方法是，把模式学习模块的执行阶段统计的模式在训练语料中的出现频率归一化处理后作为模式的先验概率，选择先验概率较高的模式作为该关系抽取依据。

### 3.3.4 关系扩展

在获取了初步的关系元组后为减少信息损失还需要对抽取的关系核心词进行扩展以及将抽取的二元关系进行多元扩展。

关系词的扩展主要存在如下情况：句子中出现和候选关系短语直接相连的动词，也应将该动词与关系短语合并，如图 3-9 所示，“宣布”和“举行”可以合并为关系短语“宣布举行”。本文在后面的描述中将单个关系词和关系短语统称为关系。

候选关系元组的扩展是指将候选关系三元组扩展为多元组。若待抽取句子中

包含词性标记为介词“p”的单词，且与其相连的名词性短语不包含在候选三元组中，则将该名词性短语加入候选关系元组中，即对候选元组进行扩展。例如下图3-13 中句子，“张文中在北京创立物美商城”，词语“在”词性标记为介词“p”，按照上述规则将其后词性标记为名词“n”的单词“北京”，加入关系元组，最终得到关系四元组(创立：张文中， 物美商城， 北京)。

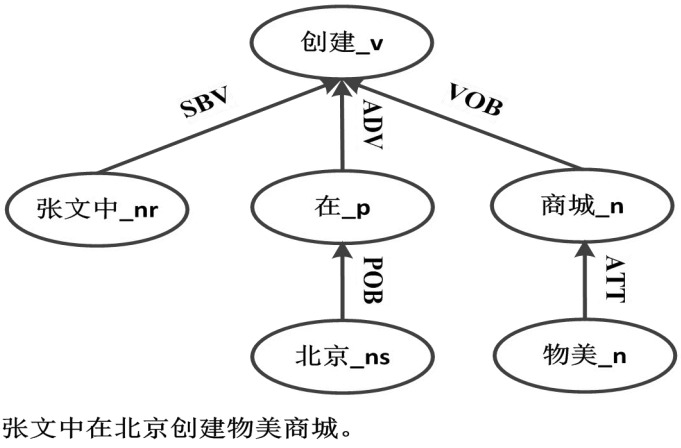


图3-13 依存分析树

## 3.4 候选关系过滤

为了提高系统的准确率，需要对上节抽取的候选关系元组进行过滤。本文采用 Logistic 对从中文语料中自动抽取出的候选关系元组进行质量评估，从而选择出其中高质量的关系元组。本方法中使用 Logistic 分类器进行质量评估，分类器中使用实体关系的统计特征以及句子的句法特征，如句子的长短、实体和关系词间的距离、实体和关系词的依存语义角色等；然后，选取一定量的开放域文本语料作为训练集，采用本文提出的候选关系抽取模块对其进行关系抽取，并采用人工判读的方式，根据关系抽取结果是否正确构造出正负样本；通过正负样本训练过滤器以确定特征权重。最后，经过参数选择和模型训练，得到最终的关系过滤模型。本文采用 5 轮交叉验证的方法训练该分类器。具体方法是将每个数据集平均分为 5 个组，每轮使用其中的一组数据为测试集，其他所有数据作为训练集，一共 5 轮，然后对 5 轮测试结果计算平均准确率，召回率以及 F1 值三个评价指标。本文通过 Wiki-500 和 Sina-500 两个数据集确定了最终逻辑斯蒂回归中所用各种特征的权值。具体方法是在 5 轮交叉验证中选择 F1 值最高的一轮的权值作为最终权重，如下表 3-2 所视。

本文在对句子结构，深层语法结构进行分析之后，最终选择了基于句法分析、基于统计以及基于句子结构三类特征共 30 个。基于统计的特征有 8 个，包含在模

式学习阶段得到的模式频率，选取该组特征的原因是模式在数据集中的频率越高说明该关系元组越常见，正确的概率有越高；基于句子结构的特征有 14 个，例如，关系词短语与名词性短语在句中的位置，该特征影响预处理结果，从而影响关系元组的正确。基于深层句法分析的特征有 8 个，主要是依存分析结果，依存句法分析是深层句法分析的一部分，和基于语句结构的特征相比，前者更能表现语句隐藏的词间关系，句法关系。具体特征见表 3-2。表中第三列为特征在 Wiki-500 上的权重。

表3-2 逻辑回归分析器输入的特征及其在Wiki-500上训练得到的权重值

| 类型 | 特征含义 | 权重 | 类型 | 特征含义 | 权重 |
|---|---|---|---|---|---|
| 基于统计 | 三元组 | -0.323 | 基于句子结构 | 名词性短语核心词和关系核心词在句子上距离小于 5 | 0.933 |
| | 四元组 | 0.001 | | 语句中关系核心词是否位于所有名词性短语片之前 | -0.194 |
| | 大于四元 | 0.021 | | 语句中关系核心词是否位于名词性短语之间 | 0.001 |
| | 匹配到的模式频数大于 1000 | -0.793 | | 语句中关系核心词是否位于所有名词性短语之后 | 0.001 |
| | 匹配到的模式频数大于 100 小于 1000 | 0.679 | | 语句中关系词在句子中是否与名词你短语相邻 | 0.003 |
| | 匹配到的模式频数大于 10 小于 100 | 0.007 | | 模式中除了关系词组还有动词 | -0.002 |
| | 匹配到的模式频数小于 10 大于 5 | -0.079 | | 关系元组中是否存在通过介词抽取得到名词性短语 | -0.198 |
| | 匹配到的模式频数小于 5 | -0.001 | 基于深层语法 | 模式对应的依存树段路径长度大于等于 3 | -0.818 |
| 基于句子结 | 语句包含的词语数大于 20 | 0.007 | | 模式对应的依存树段路径长度小于 3 | -0.003 |
| | 语句包含的词语数大于 10 小于 20 | 0.001 | | 模式中存在 SBV 和 VOB 标签 | 0.667 |
| | 语句包含词语数不大于 10 | -0.003 | | 模式中存在 ADV 和 POB 标签 | -0.449 |

| | | | | | |
|---|---|---|---|---|---|
| 构 | 句子中逗号数大于等于 2 | 0.003 | | 模式中存在 ATT 和 IC 标签 | 0.001 |
| | 句子中逗号数小于等于 1 | 0.046 | | 模式中存在 SBV 和 IC 标签 | -0.001 |
| | 名词性短语核心词核心词和关系核心词在句子上距离大于 10 | 0.005 | | 模式中存在 IC 和 VOB 标签 | 0.273 |
| | 名词性短语核心词和关系核心词在句子上距离小于 10 大于 5 | 0.400 | | 模式中存在 ROOT 和 VOB 标签 | -0.002 |

## 3.5 本章小结

本章对本文提出的中文开放式实体关系抽取算法 DPM 做了详细介绍。首先对本文提出的算法进行总体的介绍，然后分别介绍了抽取的各个模块，包括文本预处理、关系模式学习、候选关系元组抽取以及候选关系元组的过滤。

# 第四章 中文开放式实体关系抽取算法结果分析

本章主要是对本文第三章提出的开放式实体关系抽取算法DPM结果的分析。本节首先介绍了测试算法性能的数据集、评判标注，然后对实验结果进行了比较和分析。

## 4.1 数据集

为了验证本文提出的开放式关系抽取算法DPM的有效性，文章使用了四组语料进行实验，Wiki-500、Sina-500、Tecent-500 和Simple-500。在此感谢Likun Qiu等提供的Wiki-500、Sina-500 两组语料，这两组语料是其在文献中报道ZORE的实验结果时使用的测试语料，是分别从中文Wiki和Sina News中随机选取的 500 个语句。为了更客观地评估和比较本文提出的开放式关系抽取算法DPM的性能，我们从互联网上随机抽取了另外两组测试语料，Tecent-500 和Simple-500。其中，Tecent-500 是从腾讯新闻页面随机抓取的 500 条语句。经统计，Tecent-500 语料中的语句平均长度为 42.86 字。由于现实网络中人们使用的多数是简单句和短句(如微博或评论等)，因此我们进一步构造了一个由简单句构成的Simple-500 语料。Simple-500 是从新闻页面随机抓取的 500 个句子，并限定每个句子的长度在少于 40 个字，且每句中包含的逗号不超过三个。

## 4.2 评判标准

对于在本文提出的开放式关系抽取算法的评判标准继续沿用前人使用的如下几个指标：准确率 Precison，召回率Recall和综合评判指标 F1 值。

为了准确计算DPM算法在各语料库上执行关系抽取任务的结果的性能指标，本文采用专家标注的方式对四组语料中所包含的关系元组进行了人工交叉标注。两个人分别独立对每组语料中所包含的关系元组进行标注，对于二者的标记结果相同的，则认为标注结果正确，同时对二者产生歧义的标注结果进行合议，将最终形成的统一意见作为最终标记结果。经统计，双方首轮标注结果一致率达到 90%以上。本文将专家在这四个数据集上标记的所有正确关系元组记为集合S。

实验结果的获取方法如下。首先，对各个算法的抽取结果进行标注，即判断各算法得到的关系元组是否存在于正确的关系元组集合T中，如果存在，则将该元组标记True，否则标记为False。符号T表示所有标记为True的关系元组集合，符号

N表示所有标记为False的关系元组集合，符号A表示该算法抽取到的所有关系元组集合。则存在：A=T∪N。因此准确率、召回率以及F1 值计算方式如下：

$$Precision = \frac{|T|}{|A|} \times 100\% \tag{4-1}$$

$$Recall = \frac{|T|}{|S|} \times 100\% \tag{4-2}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4-3}$$

由公式 4-1 可知，准确率表示算法抽取正确的实体关系元组数量占算法抽取到的所有实体关系元组数量的百分比。准确率越高，表明算法对于该数据集进行关系抽取时，得到的正确结果的比率越高。召回率是算法抽取正确的关系元组数量占数据集中实际的正确关系元组数量的百分比。该指标反映的是算法的查全率。召回率越高，表明算法输出结果中对于该数据集而言，遗漏正确结果的可能性越低。精确率和召回率是一对具有内在矛盾的指标，通常情况下，精确率提高，召回率将降低，因此在实际应用中人们通常会在这两个指标间进行折衷，选择F1 值来客观综合评估算法的实际性能。 从公式 4-3 中可以看出，F1 值受准确率和召回率的共同影响，当二者均趋近于 1 时，F1 值也趋近于最大值 1。很明显的当F1 值越大，则说明算法综合性能越好。

## 4.3 关系抽取对比

为了验证本文提出的开放式关系抽取方法DPM的有效性，本文选择了中文开放式关系抽取中先进的算法ZORE作为比较对象。同时还针对本文提出的兼类词处理方法进行了对比实验，以证明兼类词问题的处理对提高关系抽取性能的影响。为简化描述，将对比的方法称为DPM-NoTCW，该方法与DPM几乎完全一致，但不包含兼类词处理模块。由于开放式关系抽取最后一个模块是对抽取的关系元组进行过滤，因此ZORE、DPM和DPM-NoTCW的准确率和召回率会随着逻辑斯蒂中置信度变化而变化。如果阈值的取值过小，则导致算法准确率下降，同时召回率提高；如果阈值的取值过大，则会导致算法的准确率提高，而召回率则显著下降。因此为了比较三种算法的性能，需要计算在不同置信度阈值情况下算法的准确率、召回率以及F1值。

### 4.3.1 ZORE 算法对比分析

下图展示了DPM和ZORE在四组语料中随着Logistic置信度阈值的变化而产生

的P-R曲线图。其中横坐标是算法的召回率(R)，纵坐标是算法的准确率(P)。下图4-1中每幅子图中置信度的取值范围是从0到1，每次增加0.02，在每个置信度下分别计算一次准确率和召回率，因此得到500组数据，最终得到下图的P-R曲线。

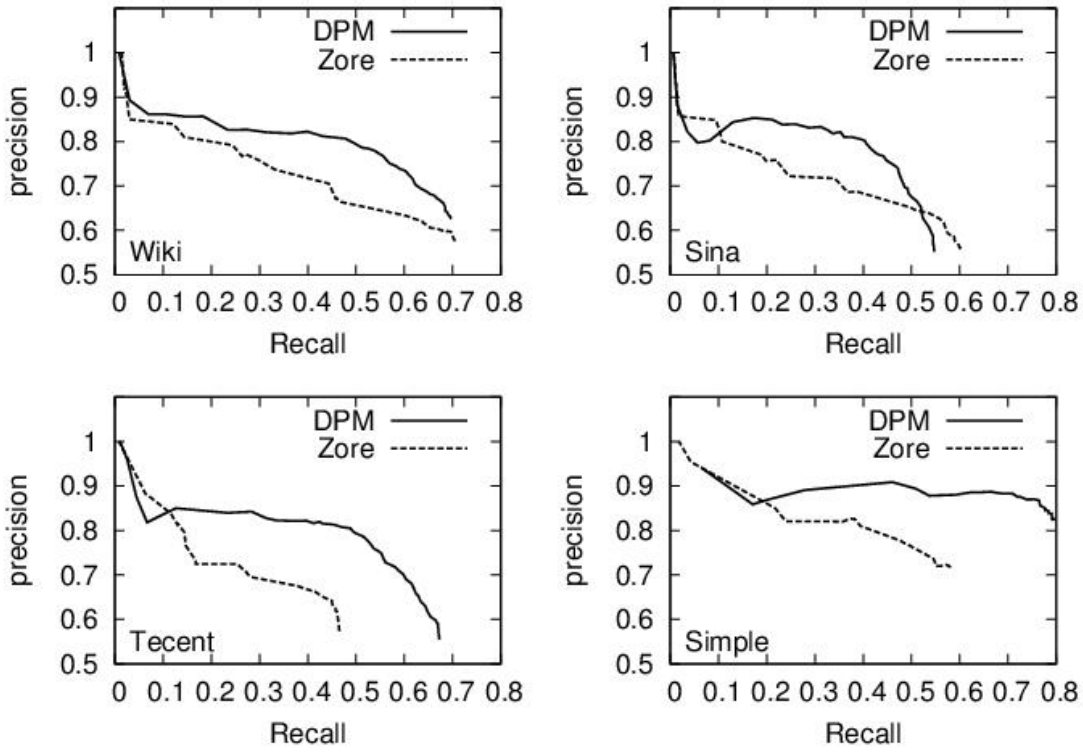


图 4-1　DPM 和 Zore P-R 曲线比较

从图中可以看出，在所有四组测试语料上，两种算法的P-R曲线均随着召回率的提高而呈下降趋势，不难看出，DPM算法的P-R曲线在绝大多数情况下位于ZORE算法的P-R曲线的右上方。这表明DPM算法的综合性能优于ZORE算法，大多数情况下当二者召回率一致时，DPM算法的结果准确率优于ZORE算法，当二者准确率一致时，DPM算法结果的召回率优于ZORE算法。特别是在Simple-500语料上，DPM算法的性能优势更为明显。当召回率在20%～80%的区间变动时，准确率一直稳定地保持在85%以上，表明DPM算法在处理短文本关系抽取任务时的性能显著优于ZORE算法，且算法的稳定性更好。主要原因是Simple-500数据结构简单且单一，因此语句中出现名词合并以及兼类词的情况较少，然而在其它三个数据集的语句中涉及名词合并以及兼类词的情况较多。

为进一步比较DPM和ZORE算法的性能，我们需要选择相同的Logistic回归阈值。本文通过在Wiki-500数据集上随着逻辑斯蒂回归阈值的变化而计算不同的F1值，最终选择阈值为0.22，此时F1值最大。因此在阈值为0.22的情况下，我们通过

比较DPM和ZORE算法的准确率、召回率以及F1值得到下图4-2结果。从下图中可以看出了DPM算法优于ZORE算法，当DPM的准确率为66%时，ZORE的准确率为60%，此时DPM和ZORE的召回率几乎相等，然而DPM的F1值明显优于ZORE。

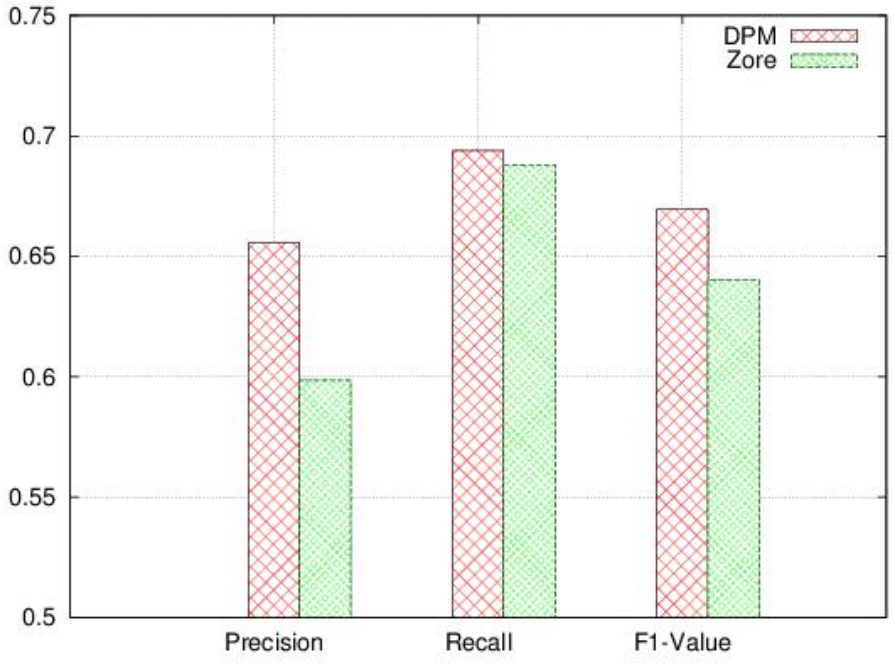


图 4-2 DPM 和 Zore 性能比较

### 4.3.2 DPM-NoTCW 算法对比分析

兼类词在中文中是很普遍的现象，如果不能很好的处理该问题，可能会对中文关系抽取产生重要的影响。因此为了验证兼类词处理在关系抽取中的重要性，本文设计了对比算法DPM-NoTCW(与DPM相比去除了兼类词处理模块其他都相同)。如下图4-3展示了DPM和DPM-NoTCW方法在Wiki-500试语料的结果。4-3图可以看出DPM算法的P-R曲线始终位于DPM-NoTCW算法的P-R曲线的右上方，这表明前者的性能优于后者。当两种算法的准确率一致时，DPM算法的P-R曲线始终位于DPM-NoTCW的P-R曲线的右方，即DPM算法的召回率高于DPM-NoTCW算法。同理当两种算法的召回率一致时，DPM算法的P-R曲线始终位于DPM-NoTCW的P-R曲线的上方，即DPM算法的准确率高于DPM-NoTCW算法。

该实验的结果表明，本文的开放式关系抽取算法中兼类词的处理是有意义的。关系抽取中兼类词的具体处理过程见3.4.1节。为了进一步验证3.4.1节中提出的兼类词处理方法的有效性，本文通过统计的方式证明本文提出的兼类词处理的有效性。具体方法是：首先从腾讯新闻数据中人工地选出300个包含兼类词的语句。然后根据人工标注出句子中存在的兼类词。这里同样使用Zpar自然语言处理工具对这

些句子进行预处理。最后，采用本文提出的兼类词处理方法处理兼类词，并人工判断处理之后的兼类词是否重新标注正确。通过观察这300个句子，可以发现其中有78个句子中的兼类词满足方法一的条件，135个句子满足方法二的条件。根据本文提出的两种兼类词处理办法对名词性短语合并的平均纠错正确率达到63%。

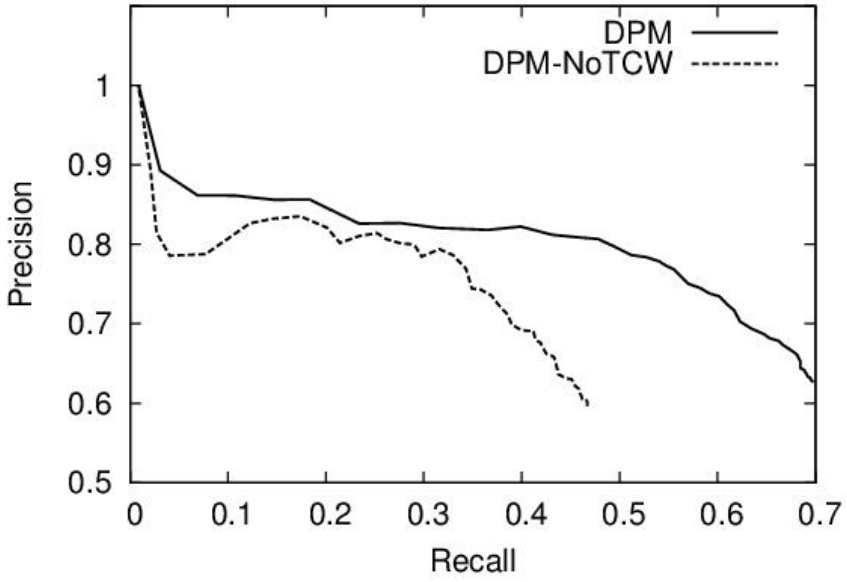


图4-3 DPM和DPM-NoTCW PR曲线比较

## 4.4 模式评估

本文利用训练语料学习得到3769个关系模式，经过聚类处理最终得到2431个关系模式，其中有621个高频率的关系模式。如下图4-4展示模式的频率分布图，其中横坐标表示第三章中学习得到模式的频率，纵坐标表示在该频率下模式数量。为了作图的美观和客观性，图中去除了特别离散的点如模式“IC(v):SBV(n) ; VOB(n)”其频率为7059。图中的趋势符合人类认知，模式的频率越高在该频率下的模式数量就越少，也就是说现实生活中非常明显和常见的模式是相对较少的。

为了评估模式的准确性，我们设置两组实验数据。第一组是从第三章中学习得到关系模式中选择频率最高的100个关系模式，第二组从中随机选择100个关系模式。采用人工判别的方法，实验使邀请两位专家分别独立对这两组模式进行判断。对于二者的标记结果相同的，则认为标注结果正确，同时对二者产生歧义的标注结果进行合议，将最终形成的统一意见作为最终标记结果。为方便两位专家进行判断，我们提供的是关系模式以及关系模式对应的训练语料。最终的结果是第一组高频率的关系模式的准确率为95%，第二组随机抽取的关系模式的准确率为

91%。其中主要的错误是其中抽取的较长的关系模式，如IC(v):SBV(v),ATT(n)# ADV(p), POB(n)#IC(v), IC(v),IC(v), ADV(t)。

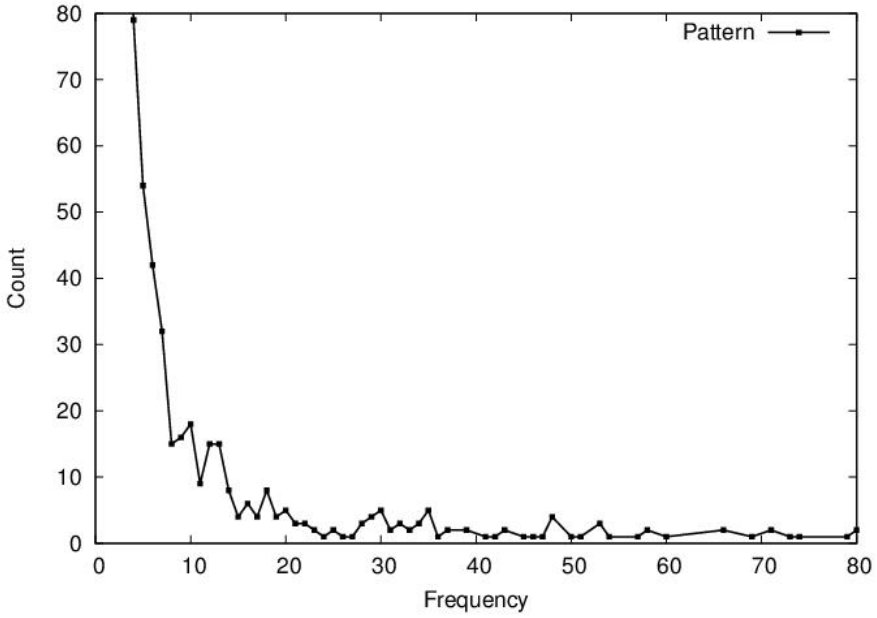


图4-4 模式分布图

## 4.5 总结

本章通过将本文提出的开放式关系抽取算法DPM以及DPM-NoTCW和前人提出的算法ZORE在Wiki-500、Sina-500、Tecent-500 和Simple-500 四个数据集进行比较。充分证明了本文提出的算法DPM的有效性，同时分析了本文学习得到的关系模式的准确性。

# 第五章 中文实体关系抽取原型系统实现

## 5.1 系统框架

本文在实现第三章提出的中文开放式关系抽取算法DPM的同时，利用百科网页抽取上下位关系和属性关系，并利用两种方式抽取的高质量关系元组构建关系知识库。

### 5.1.1 系统功能模块

由图 5-1 所示，本章关系抽取原型系统主要分为四大功能模块，分别为数据爬取和解析模块、数据预处理模块、百科网页关系抽取模块、开放式关系抽取模块。下文分别对不同的模块进行功能描述以及简单的介绍。

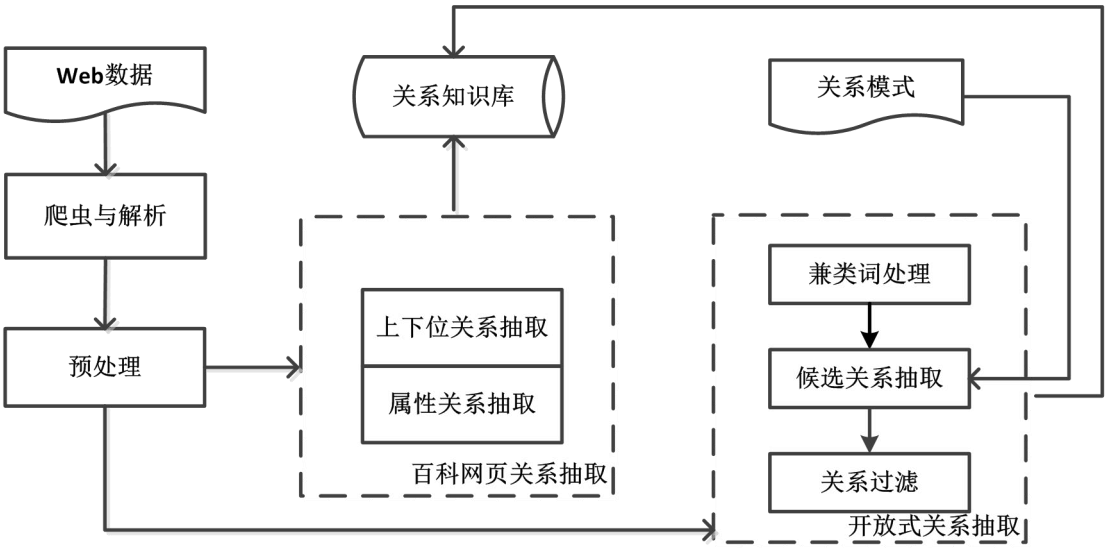


图 5-1 中文关系抽取系统结构图

数据爬取和解析模块。本模块首先通过自动的方式抓取网络百科、垂直网站或者其他的网络文本数据，其次通过解析得到命名实体加入到自然语言处理工具中提高处理的准确性，再者通过解析百科页面中相关信息如信息框等作为关系元组的重要来源，最后通过解析获取文本数据作为构建开放式关系抽取的重要数据来源。

数据预处理模块。本模块主要包括如下两个主要功能：首先读取数据爬取和解析模块的数据，对数据进行分词、词性标注和依存分析基本数据预处理操作，然后对自然语言处理之后的输出结果去除其中的停用词。

百科网页关系抽取模块。本模块主要是对爬取的百科信息进行解析处理得到上下位关系和百科属性关系，并存入关系知识库中。

开放式关系抽取模块。本模块主要是对数据预处理后的文本数据利用 3.3 节学习得到的关系模式抽取候选关系元组，然后利用 3.4 节介绍的Logistic回归模型对抽取的候选关系元组进行过滤。同时也将本模块抽取的高质量的关系元组加入到关系知识库中。

### 5.1.2 系统流程

本文基于简单实用的原则设计实现了中文关系抽取系统。本系统主要采用两种方法抽取关系元组。第一种是基于网络百科数据抽取其属性框中丰富的实体属性关系，以及百科数据中上下位关系。此外，本系统采用本文实现的开放式关系

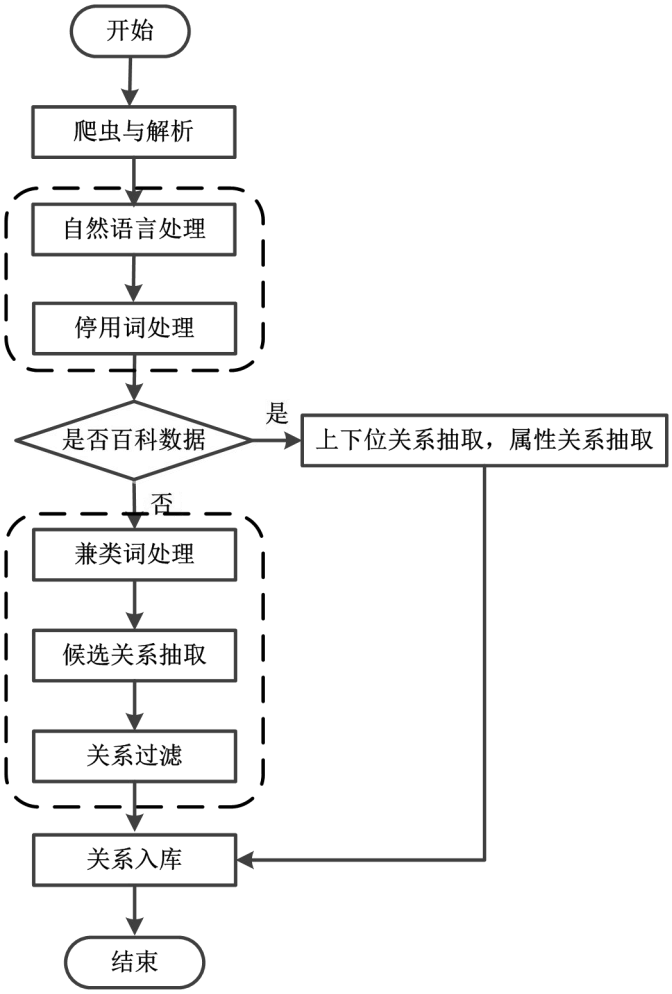


图 5-2 中文关系抽取系统流程图

抽取算法抽取更为丰富的实体关系元组。系统的流程如下 5-2 所示，首先通过爬虫

与解析模块对 Web 数据（主要包括百科数据和需要进行开放式关系抽取的网页数据）进行爬取，爬取之后将百科网页数据的源码直接以文件的形式存储在本地。将需要进行开放式关系抽取的网页数据，使用 Jsoup 对爬取的网页源码进行解析然后将解析结果以文件的形式存储到本地。其次，对解析得到的文本数据进行分句、分词、词性标注、依存语义分析以及对其中的停用词进行处理。再次，对获取的百科开放分类和词条标签抽取上下位关系，同时抽取百科页面信息框中属性关系元组，并将抽取的关系元组存入关系知识库中。最后为丰富关系知识库中的关系元组，系统使用第三章学习得到的关系模式对预处理后的文本抽取实体关系元组。

## 5.2 系统开发环境及相关工具

### 5.2.1 系统开发环境

(1) 操作系统：Windows7，系统使用 JAVA（jdk1.7.0_75）语言开发，IDE 开发工具使用 Eclipse。

(2) 硬件环境：PC 机，处理器 inter i5 CPU2.6G Hz，内存 8G，硬盘 500G。

表 5-1　Zpar依存分析标签含义

| 序号 | 标签 | 含义 | 序号 | 标签 | 含义 |
|---|---|---|---|---|---|
| 1 | ROOT | 核心关系 | 15 | RAD | 右附加关系 |
| 2 | SBV | 主谓关系 | 16 | RADC | 非共享右附加关系 |
| 3 | VOB | 动宾关系 | 17 | PUS | 句中标点 |
| 4 | IOB | 间宾关系 | 18 | PUN | 句末标点 |
| 5 | POB | 介词宾语 | 19 | TPC | 主题 |
| 6 | ACT | 动词性宾语 | 20 | VV | 串行动词 |
| 7 | ATT | 定中关系 | 21 | MT | 时态 |
| 8 | ADV | 状中关系 | 22 | NUM | 数量 |
| 9 | CMP | 动补关系 | 23 | QUN | 度量关系 |
| 10 | APP | 同位语关系 | 24 | QUC | 前置量词 |
| 11 | COO | 并列关系 | 25 | QUCC | 非共享前置量词 |
| 12 | COS | 右共享并列关系 | 26 | ISC | 非共享独立结构 |
| 13 | IC | 独立子句 | 27 | LAD | 左附加关系 |
| 14 | IS | 独立结构 | 28 | RED | 重复元素 |

### 5.2.2 系统使用的相关工具

(1) Zpar

Zpar是一个自然语言处理工具，其可以进行分词、词性标注、依存分析。Zpar自然语言处理工具可以支持多种语言以及多种语法形式。其中汉语和英语是最着重处理的，同时对其他部分语言的也提供支持。例如Zpar也支持罗马尼亚语言。就语法形式而言，Zpar支持上下文无关文法(CFG)，依存句法和组合范畴语法(CCG)。

Zpar词性标注是基于北京大学的人民语料库和其词性标注体系。由于北京大学的词性标注体系过于复杂（词性标注体系超过 100），Zpar将其简化为 33 个标注体系。Zpar的依存分析标签说明如上表 5-1 所示，词性标注体系标签说明如下表 5-2 所示。

表 5-2 Zpar词性标注标签含义

| 序号 | 标签 | 含义 | 序号 | 标签 | 含义 |
|---|---|---|---|---|---|
| 1 | a | 形容词 | 18 | b | 区别词 |
| 2 | d | 副词 | 19 | m | 数词 |
| 3 | c | 连词 | 20 | j | 简称缩写 |
| 4 | e | 感叹词 | 21 | k | 后缀成分 |
| 5 | f | 方位词 | 22 | l | 习语 |
| 6 | o | 拟声词 | 23 | n | 名词 |
| 7 | g | 语素词 | 24 | nr | 人名 |
| 8 | h | 前缀成分 | 25 | ns | 地名 |
| 9 | i | 成语 | 26 | nt | 机构团体 |
| 10 | p | 介词 | 27 | nz | 其他专名词 |
| 11 | q | 量词 | 28 | r | 代词 |
| 12 | s | 所处词 | 29 | x | 非语素词 |
| 13 | y | 语气词 | 30 | z | 状态形容词 |
| 14 | t | 时间词 | 31 | u | 助词 |
| 15 | v | 动词 | 32 | w | 标点 |
| 16 | nrf | 姓氏 | 33 | nrg | 姓 |
| 17 | nx | 非汉语名词 | | | |

(2) Jsoup

无论采用何种方式从网络爬取源码，对爬取的网页源码进行解析得到符合需求的数据这一步骤是不可避免的。此处所指的页面解析是指从 HTML 网页源码解析得符合需求的数据。本文在数据爬取和解析模块的过程中使用 Jsoup。其是一款基于 JAVA 的 HTML 解析器，可直接解析 URL 网页地址或者 HTML 的网页源码。它提供了一套方便使用的 JAVA API，可通过 DOM，CSS 以及类似于 JQuery 的操作方法来解析得到网页文本数据。Jsoup 使用十分方便简单，类似于 JavaScript 操作页面 DOM 对象，对已有 JavaScript 应用经验的开发人员来说很直观、方便使用。Jsoup 的使用主要包括如下两个步骤。步骤一加载 HTML 页面，生成 Document 对象实例，其可以通过如下三种方式进行加载：根据给定的 HTML 格式的字符串直接加载文档、根据本地 HTML 文件加载文档或者根据给定的 URL 地址加载文档。步骤二通过解析 HTML 标签以获取相关的文本数据。

(3) Neo4j

图数据库利用相关的图形理论存储实体信息以及实体与实体间的关系信息，其是一种非关系型数据库。由于查询过程缓慢、时间代价过大，关系型数据库存储实体间关系数据的效果并不理想。实体关系数据的形式就是点和点直接的图关系，因此图数据库的这一特点非常适合存储该类型的数据，弥补了关系数据库的缺点。现在比较流行的图数据库主要有Neo4j、FlockDB、AllegroGraph、GraphDB和InfiniteGraph等，本文使用Neo4j对抽取的关系元组进行存储构建知识库。

Neo4j是一个高性能的开源图形数据库，以图的形式存储结构化数据。Neo4j可以被认为是高性能的图引擎，其性能可以和成熟数据库所媲美。Neo4j具有如下特点：首先，其是一个面向对象的、灵活的图结构，而不是严格、静态的表；其次Neo4j具有完备的的事务特性；再者，Neo4j是基于JAVA实现的，同时也支持Ruby和Python等其他编程语言。

Neo4j是以节点和边(关系)模式进行存储的，其中结点用Node类表示，边用 *Relationship* 类表示，边即关系的类别使用 *RelationshipType* 类(关系的类别需要预先定义)或者 *DynamicRelationshipType* 类(关系的类别不需要预先定义)表示。每个节点可以包含各种属性， *setProperty*() 方法是Node类中的一个方法，该方法能够实现对节点信息的保存。若要实现多个结点之间的关系网，可以使用 *createRelationshipTo*() 方法。此外对于 *Relationship* 关系对象可以设置其属性，关系属性可以描叙为两个节点间的关系类型，如可以设置 “张三”和“李四”两个结点的 *Relationship* 属性为同事、同学或者校友。这种存储方式可以方便的查找满足某种关系类型的结点。

## 5.3 数据爬取和解析模块

### 5.3.1 数据爬取和解析

本模块的目的是从网络获取相关文本数据，包括实体数据、百科页面数据以及用于抽取关系元组的纯文本数据。本模块主要分为如下三个部分。第一部分是数据爬取也就是获取网页源码。第二部分是对获取的网页源码就行解析得到其中的URL列表。URL的获取，主要负责提取网页中的符合一定条件的URL，并将解析得到的URL信息存入文件中供进一步抽取处理。第三个部分是文本数据的解析,本部分主要是对网页源码进行解析，获取得到符合需求的文本数据同时将获取的文本数据存入文件中。如下图 5-3 为本模块流程图。

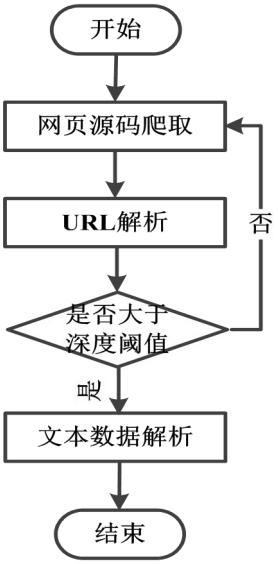


图 5-3 数据爬取和解析流程图

本文使用JAVA Net包中的类获取网页源码，然后使用Jsoup解析得到的符合条件的URL列表和需要的网页文本数据。

URL解析是依据输入的URL根节点地址获取的网页源码，然后在该源码中找出符合一定条件的所有URL列表。根据根节点搜寻子节点列表的方法主要有深度优先和广度优先搜寻。例如本文对百度组织机构数据，获取URL的整体过程是先根据根节点URL（百度开放分类http://baike.baidu.com/class/795.html），进行页面解析从中选择符合条件的机构类的URL，如图 3-8 中选出的黑色框内。接下来依次进入到各个子类别中获取对应类别的URL集合。

本文从百度机构开放分类共 19 个大类中大约获取 158000 个URL。在获取所有URL后，然后开始爬取各个URL对应的网页源码并使用Jsoup解析获取的网页源码中符合要求的数据，并将获取的数据进行存储。需要注意的是本文使用是根据

页面的层级结构来逐层爬取URL的方法。

### 5.3.2 重要数据结构和功能函数

```
class   UrlInfo{
    private String url;
    private int urlLevel;
}
class   WebCodeAndURL{
    String urlPath; //需要爬取的网页链接
    public String getCharset(String urlString): //得到网页编码方式
    public String getWebCode(String ulrString,String charset): //得到网页源码
    public ArrayList<UrlInfo> geturl(String urlString): //得到网页中包含的URL集合
}
```

由于网络中不同网页的编码方式可能不同，因此为了准确获得网页源码首先需要获得对应网页的编码方式。本系统中 getCharset(String urlString)可以获取对应网页的编码方式。在得到网页的编码方式之后，getWebCode(String ulrString,String charset)方法可以进一步获取该网页的源码。最后使用 ArrayList<UrlInfo> geturl (String urlString)方法获取网页中包好的 URL 集合，其中获取的 URL 信息存储在 UrlInfo 数据结构中，其中主要包含 URL 链接以及 URL 对应的级别。在获取了更多的 URL 集合之后可以进一步使用 getCharset(String urlString)和 getWebCode (String ulrString,String charset)方法获取更多的网页信息。

### 5.3.3 输入输出

输入：String url 初始的网页链接

输出：String webCode 获取的网页源码，ArrayList<UrlInfo> urlInfoList 存放网页中获取的 URL 信息，其中包含 URL 链接以及 URL 对应的级别。

## 5.4 文本预处理模块

本模块主要对待处理的文本数据利用自然语言处理工具将其转换成带有自然语言处理标记的语句集合并去除其中无意义的词汇。本模块主要包括自然语言处理和去除停用词两个步骤。

### 5.4.1 自然语言处理

分词、词性标注以及依存分析都是自然语言处理研究的最重要和最基本的问题，都涉及复杂的算法。虽然目前还没有能被完美地解决，但是利用一些现有的自然语言处理工具，其文本标注的质量还是可以接受。通过比较中科院的ICTCLAS 系统 、复旦大学的Fudan NLP系统、哈尔滨工业大学社会计算与信息检索研究中心发布的 LTP 自然语言处理平台、斯坦福大学的Stanford-Parser以及Zpar等，本文最终使用效率和准确率相对较高的Zpar对网页正文进行分词、词性标注以及依存分析。另一方面，当然我们也意识到了自然语言处理工具的限制性，我们利用抽取构建的关系知识库中的实体作为自然语言处理的字典，以减小自然语言处理导致的错误。如图 3-8 是“奥巴马总统与中国驻美国大使进行会谈”的分词、词性标注和依存分析结果。其中每个圈中的词汇是分词的结果；词汇后面的英文字母如“n”、“p”、“ns”等是对应词汇的词性；图中的有方向的线表示词汇和词汇之间存在依存关系，线上字母如“SBV”、“ATT”、“VOB”等表示词汇和词汇之间的依存关系类型。

### 5.4.2 停用词处理

没有太大实际意义但出现频率却相对较高的词汇称为停用词，例如“的”、“把”、“了”等。由于文档中高频率的停用词会减少有实际意义词汇的作用，因此其对关系抽取任务会产生一定的消极作用。因此，在实际关系抽取过程中，停用词必须被剔除。一般地，为了将待处理文本的停用词去掉，引入停用词表过滤停用词是当前有效且方便的方式。目前存在一些较好的停用词表能够直接利用而无需自己统计。本文使用的停用词表是直接从互联网获取的，本文是采用先分词后去除停用词的方式。

### 5.4.3 重要数据结构和功能函数

```
class DPTreeNode {
    private String nodeID;//依存分析树节点编号
    private String nodeContent;//依存分析树节点词语
    private String nodeTag;//依存分析树节点词语词性标签
    private String parentNodeID;//依存分析树节点的相连结点编号
    private String dpEdge;//依存分析树中结点与父节点间的依存关系标签
    }
class NLPProcessing{
```

String sentence;

public String tagSen(String sentence)://对语句进行分词和词性标注

public String removeStopWord(String posSentence,ArrayList<String> stopWord):

//对分词和词性标注后的结果去除停用词

public DPTreeNode parseSentence(posSentence)://依存分析

}

系统中使用 DPTreeNode 数据结构存储依存分析的结果，其中 nodeID 表示依存分析树节点的编号，nodeContent 表示依存分析树节点词语，nodeTag 表示依存分析树节点词语词性标签，parentNodeID 表示依存分析树节点的相连结点编号，dpEdge 表示依存分析树中结点与父节点间的依存关系标签。系统使用 ArrayList<String> stopWord 存储停用词集合。tagSen(String sentence)方法对输入的语句进行分词和词性标注，然后使用 removeStopWord(String posSentence, ArrayList <String> stopWord)方法对分词和词性标注后的结果去除停用词，最后使用 DPTreeNode parseSentence(posSentence)方法对去除停用词后的结果进行依存分析。

### 5.4.4 输入输出

输入：String sentence 爬虫和解析后得到的语句，ArrayList<String> stopWord 停用词表。

输出：DPTreeNode 语句返回的依存分析结果。

## 5.5 百科网页关系抽取模块

百度百科是一个开放的在线百科全书,其中包含了海量人类知识以及语义关系。此类页面的特点是其内容都是某一领域的权威人士撰写与更新，而且随着词条版本的不断变更，其内容通常也变得更加精确与完善。重要的是，百科类网站对热点词和流行词反应敏感,这些词可以在较短的时间内被加入百科页面中，这对于扩展知识库的内容具有重要作用。本节从知识库建设方面出发,研究了百科页面信息抽取方面的处理方法。本节基于百度百科开放分类和词条标签抽取出反映实体层次的上下位关系，同时利用信息框抽取了属性关系元组。

### 5.5.1 上下位关系抽取

实体的上下位关系抽取是构建知识库中分类体系的关键，同时为丰富关系知识库内容起着重要的作用，目前该领域吸引了大量研究者。该关系类型主要是用于描述实体间包含与被包含的语义关系。概念上外延更广的实体一般称为上位词，

概念上内涵更窄的实体一般称为下位词，也就是说下位词是上位词的实例或者子类。如“交通工具”的下位词包括“汽车”、“飞机”、“奥迪”、“宝马”，而同时“奥迪”和“宝马”也是“汽车”的下位词。因此,同一个上位词的下位词之间也可能存在上下位关系,这就构成了上下位关系中的层次结构。多个层次结构也就是构成了实体之间的树状结构，相邻实体之间具有严格的 ISA 关系，也就是下位词共享上位词的性质。

本文通过利用众多类似图 3-10 的“社会”开放分类抽取实体之间的上下位关系，如“人物”、“自然”等开放分类。同时本文还利用百科页面的词条标签抽取得到实体的上位词，及获得了相关的上下位关系。例如下图 5-4 为“奥巴马”百科页面，从中可以抽取得到（奥巴马, ISA, 政治人物）、（奥巴马, ISA, 人物）、（奥巴马, ISA, 元首）等上下位关系。从本例中可以看出一个实体可以具有多个上位词,这样我们抽取的上下位关系更加丰富。大大方便了某些基于知识库的查询任务，例如当我们分别查询知识库中所有的“政治人物”或者所有的“元首”，知识库会直接给出相应类别的所有下位词，不需要在整个知识库中搜索答案，极大的节约了时间。

### 5.5.2 属性关系抽取

由于网络百科页面中的信息框中包含大量属性关系元组，这为构建知识库以及构建用于学习关系模式的语料提供了重要的基础，其中属性关系元组的形式为(实体，属性名，属性值)。实体就是当前百科网页的实体对象,属性名和属性值是百科信息框中的主要成分,所以生成属性关系元组的工作重点就是从百科页面的信息框中抽取属性名和对应的属性值。信息框中的属性值存在如下两种情况。第一种情况是属性名和属性值是一对一的关系，此时可以直接解析得到属性关系三元组。第二种情况是属性名和属性值是一对多的关系，此时如果直接抽取得到属性关系三元组，其对 3.3.1 节中训练语料的获取以及关系元组的可用性产生一定的影响。第二种情况时,属性值通常由标点符号隔开,如果在 3.3.1 节中直接在搜索引擎中搜索未处理的属性关系元组,其得到的结果中会遗漏很多相关文本信息,进而导致用于抽取关系模式的语料减少而且质量降低,因此需要对多值属性进一步细化,进而生成多个新的细化的属性关系元组。如图 5-4 所示，“贝拉克奥巴马”的百度百科页面,其中信息框中的“中文名”、“外文名”、“国籍”等是一对一关系，“职业”、“毕业院校”、“代表作品”等是一对多关系。

本模块主要分为如下三个步骤：属性关系初步抽取模块，主要是爬取百科网页源码并初步解析出属性关系三元组；属性切分模块，主要是判断抽取的属性值


图 5-4 “奥巴马”百科页面

中是否是一对多的多属性值的属性关系，如果存在就对该属性进行切分得到多个属性关系三元组；将抽取的关系元组加入到实体关系知识库中。流程图如下 5-5 所示。具体的过程是，首先通过 5.3 中的数据爬取和解析模块初步得到信息框中属性关系元组；然后观察抽取的信息框中是否存在一对多的属性关系。如果某个属性值中存在“。”、“,”、“，”、“、”、“；”、“;”、“：”、“:”以及同时有多个书名号时，则认为改属性是一对多的属性关系。因此在抽取属性关

系三元组的时候需要将其切分为多个属性关系三元组。例如上图中的“职业”属性是一对多的属性关系，因此通过切分处理得到如下三个属性关系三元组(奥巴马，职业，政治家)、(奥巴马，职业，律师)和(奥巴马，职业，总统)。最后将抽取的属性关系三元组存入关系知识库中。

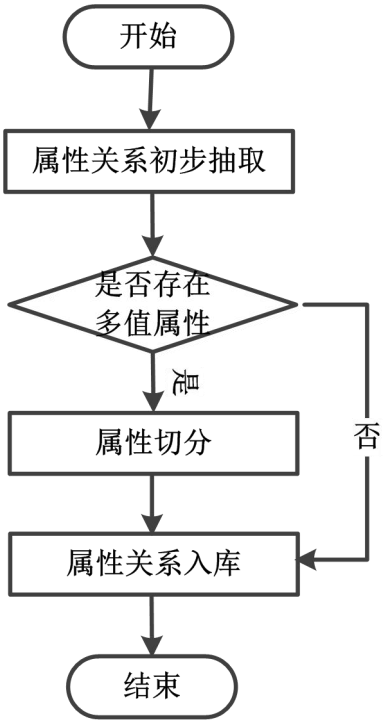


图 5-5 属性关系抽取流程图

### 5.5.3 重要数据结构和功能函数

本模块利用 Neo4j 中三个重要的数据结构 Node（表示关系知识库中的结点），Relationship（表示关系知识库中的结点间关系），GraphDatabaseService（表示关系知识库）存储抽取的上下位关系和属性关系元组。

public GraphDatabaseService openClassExtra (String htmlStr):本函数主要的功能是从开放分类中抽取上下位关系并存入关系知识库中。

public GraphDatabaseService openTagExtra (String htmlStr):本函数主要的功能是从词条标签中抽取上下位关系并存入关系知识库中。

public GraphDatabaseService infoboxExtra (String htmlStr):由于百科信息框中含有丰富的属性关系元组，因此设计和实现了该函数从信息框中抽取属性关系并存入关系知识库中。

### 5.5.4 输入输出

输入：String htmlStr 爬虫获取的网页源码。

输出：GraphDatabaseService 本模块抽取的上下位关系和属性关系知识库，其中包含各种实体结点以及实体间的关系。

## 5.6 开放式关系抽取模块

### 5.6.1 开放式关系抽取

由于百科网页信息框中的属性关系元组是有限的，还有大量的实体关系数据隐藏在非结构化的公开数据中。因此本模块将实现第三章中文开放式实体关系抽取算法DPM，从非结构化的纯文本中抽取实体关系元组，以拟补百科网页中抽取的关系元组不足的问题，进而丰富关系知识库。具体流程如 5-6 图。其主要过程如下：首先，对预处理后(自然语言处理和去停用词)的文本数据进行兼类词的处理；其次，使用 3.3 节中学习得到的高质量的关系模式抽取候选关系元组；再次，对抽取的候选实体关系元组进一步使用机器学习算法进行过滤得到高质量的关系元组；最后，将抽取的高质量的关系存储在关系知识库中。在前面的第三章已经对下图的各个模块进行了详细的介绍，本节就不再作详细的解释。

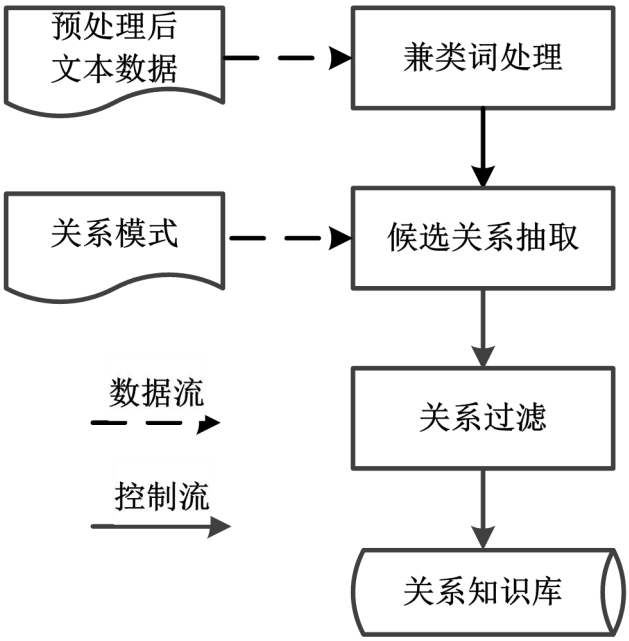


图 5-6 开放式关系抽取结构图

### 5.6.2 重要数据结构和功能函数

本模块中系统利用第三章中学习得到的关系模式Pattern抽取关系元组，其数据结构如下，主要包括模式中的关系词特征，模式中每个实体到关系词的依存路径特征，以及该模式的频次。

```
class Pattern {
    private String relationTag; //该模式的关系词特征
    private List<List<String>> arg2RelationPath;//模式中实体到关系词依存路径
    Private int frequency;//pattern 的频次
}
```

如下数据结构 RelationTuple 表示抽取的关系元组，包括关系词，关系元组的实体列表，产生该关系元组的关系模式以及用于计算该关系元组置信度相关特征。

```
class RelationTuple {
    private String realtionWord;//此关系元组的关系词组
    private List<String> entityList;//此关系元组的所有实体列表
    private Pattern pattern;//产生词关系元组的pattern
    private List<String> feature;//此关系元组和其语句用于关系过滤的相关特征
}
```

如下数据结构 ExtractResult 为每个语句对应开放式关系抽取的结果，其中包含用于抽取关系语句，每个语句抽取的一个或者多个关系元组（使用上段中的 RelationTuple 表示），以及此语句的依存分析树。

```
class ExtractResult {
    private String sentence; //执行抽取的句子
    private List<RelationTuple> relationTuples;//抽取到的关系元组
    private List<DPTreeNode> DPTree; //此句子的依存分析树
}
```

如下数据结构 FinnalReturnType 为本模块最终得到的结果，其中包含每个关系元组以及关系元组对应的置信度。

```
class FinnalReturnType {
    private List<Double>    confidence;//每个语句抽取的关系元组的置信度
    private List<ArrayList<String>>    tuple;//每个语句抽取的关系元组
}
```

如下为主要的功能函数：

public List<DPTreeNode> detectTC(List<DPTreeNode> DPTree):该函数的主要功能

是对预处理之后的结构进行兼类词处理。

public List< Pattern > findBestMathing (List<Pattern> patterns,List<DPTreeNode> DPTree,List<List<String>> NPgroup)：在兼类词处理之后，该函数实现了在模式集合中寻找与语句匹配的关系模式。

public ExtractResult doExtract(String sentence, List<DPTreeNode> DPTree, List <Pattern> patterns):该函数使用寻找得到的合适的关系模式抽取关系元组。

public List<String> featureConstruct(RelationTuple relationTuples, String sentence, List<DPTreeNode> DPTree): 该函数实现了用于计算该关系元组置信度相关特征。

public FinnalReturnType predictFun(ArrayList< ExtractResult > allPredict,double[] feaWeight):该函数的主要功能是对每个语句抽取的关系元组进行置信度计算。

public GraphDatabaseService save2GraphDB(FinnalReturnType relationTuples): 本函数主要是将抽取的关系元组存储到图数据库 Neo4j 中。

### 5.6.3 输入输出

输入：Pattern 存储第三章获取的关系模式，其中 relationTag 表示关系词的词性和依存语义分析标签，arg2RelationPath 表示实体到关系词间的依存关系路径，frequency 表示该模式的频次。String sentence 表示待抽取关系元组的语句，List<DPTreeNod>DPTree 表示待抽取语句的依存分析结果。

输出：FinnalReturnType 表示开放式关系抽取返回的结果，其中 tuple 表示抽取的关系元组，confidence 表示关系元组的置信度。最终将其中的高置信度的 tuple 存储到关系知识库 GraphDatabaseService 中。

## 5.7 系统测试和关系知识库展示

本节将对本文实现的中文关系抽取系统作简单的介绍。主要介绍和测试了原型系统实现的几个主要功能：词性标注、依存分析、属性关系抽取、开放式关系抽取以及关系知识库中数据的展示。

### 5.7.1 词性标注测试

本节主要是对输入的文本进行分析处理，输出分词和词性标注结果，具体如图 5-7 所示。本文没有构建自己的分词和词性标注工具，而是使用效率和性能较高的公开自然语言处理工具Zpar。为了提高分词的准确率，本文在原型系统中加入了自己的分词字典库，此外兼类词的处理也进一步提高了词性标注的准确率。

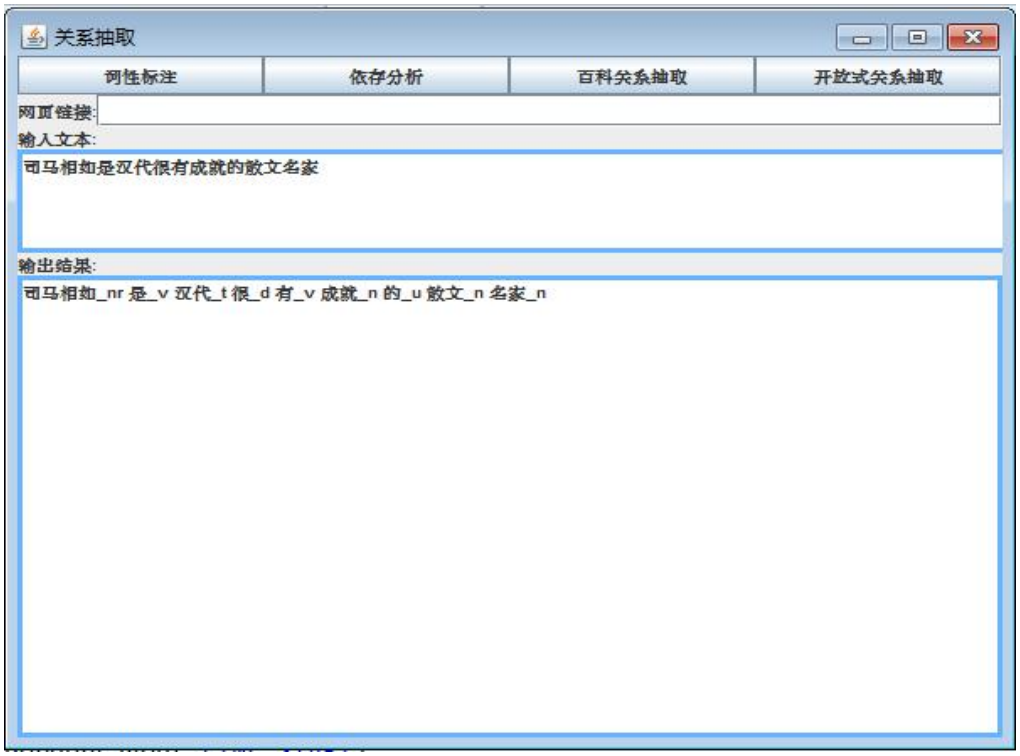


图 5-7 词性标注功能测试

## 5.7.2 依存分析测试

本节主要是对输入文本进行分析处理，输出语句的依存语义分析结果，具体


图 5-8 依存语义分析功能测试

如图 5-8 所示。本文没有自己构建依存分析工具，而是使用性能和效率相对较高的公开自然语言处理工具Zpar。此外，系统中兼类词的处理，进一步提高了依存语义分析的准确性。

### 5.7.3 百科属性关系抽取测试

中文百科属性关系抽取实现的功能是在文本框中输入某个实体的百科网页，输出的结果是从该实体的百科网页中抽取的属性关系元组以及实体的上下位关系。其中对于多值的属性关系已经进行切分处理，此外百科网页中实体的分类标签构成的ISA关系也已经抽取。如下图 5-9 为“马云”百科页面的抽取结果。


图 5-9 百科属性关系抽取功能测试

### 5.7.4 开放式关系抽取测试

中文开放式关系抽取功能模块实现的功能是在文本框中输入待抽取的纯文本，输出的结果是从纯文本中抽取的实体关系元组，如图 5-10 所示。

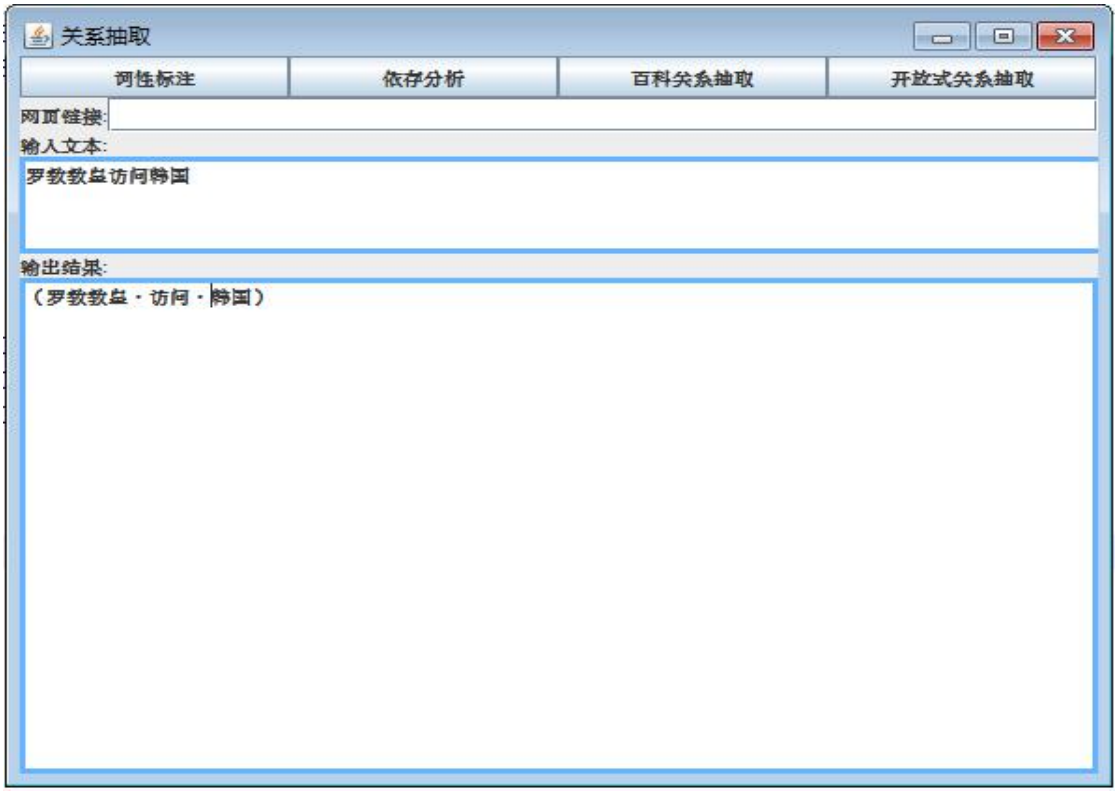


图 5-10 开放式关系抽取功能测试

### 5.7.5 关系知识库展示

本文无论是从百科网页中抽取的属性关系三元组还是利用开放式关系抽取算

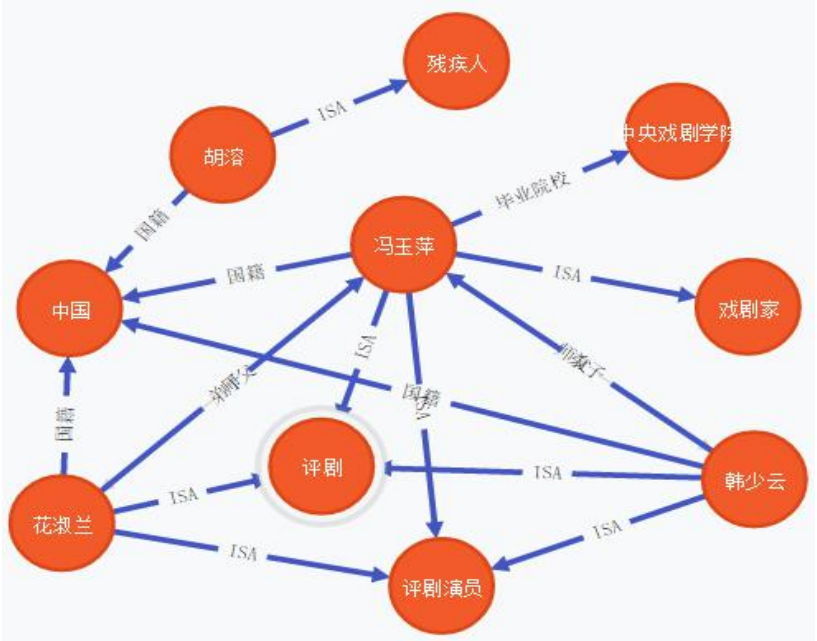


图 5-11 关系知识库展示

法抽取的关系三元组最终都存储在图数据库 Neo4j 中。最终构建了包括 814784 实体结点，3259209 个关系的关系知识库。由于系统不断的从文本中抽取实体关系元组，因此关系知识库中实体和关系不断的增加。

上图 5-11 展示了本文抽取的关系知识库中部分实体关系元组。其中每个单独的结点表示实体，实体间的连线表示实体与实体间存在的关系。如图中的“冯玉萍”和“中国”之间存在一条“国籍”的连线，也就是存在(冯玉萍，国籍，中国)这样一个关系元组。

## 5.8 总结

本章主要介绍了中文实体关系抽取系统的设计与实现。首先整体介绍了本系统的功能模块，然后介绍了在实现系统时用到的相关的工具和技术，其次详细介绍了系统的几个功能模块，其中包括数据爬取和解析模块、百科网页关系抽取模块和开放式关系抽取模块，最后对相关功能模块进行测试，测试结果表明系统各个功能模块运行正常。

# 第六章 结束语

随着网络的快速发展，互联网已经成为现今蕴含信息量最大的平台。然而对于人类来说，如何从如此浩大的数据中快速准确地获取信息和知识是个重大的挑战。因此如何将网络中海量的无结构化、难以快速阅读的数据变成结构化、能够快速阅读的数据越来越重要。现在对于英文的开放式抽取研究相对较多且取得一定成果，但是对于中文的研究相对较少。本文提出了一种从中文文本中抽取结构化数据的开放式关系抽取算法 DPM。此外，本文还实现了中文关系抽取系统，该系统可以从百科网页抽取上下位关系、实体属性关系以及使用开放式关系抽取算法从纯文本中抽取关系元组以丰富关系知识库。

## 6.1 全文总结

本文通过学习现有的开放式关系抽取算法，并了解现有算法的核心理论，发现前人关系抽取算法的不足以及抽取过程中忽略的问题。据此本文提出了一种中文开放式关系抽取算法解决前人忽略的问题，改善了中文开放式关系抽取算法的性能。此外，本文实现了中文关系抽取系统，系统首先抽了百科网页中的上下位关系和实体属性关系。为了进一步丰富关系知识库中的关系元组，系统使用本文提出的开放式实体关系抽取算法 DPM 从纯文本中抽取关系元组。概括起来论文的主要贡献如下：

(1) 本文提出了一种基于兼类词句法角色统计规律的兼类词处理方法，而且通过实验证明了该方法能够有效地改善兼类词问题，进一步提高了中文开放式关系抽取算法的性能。

(2) 本文提出了一种利用词法分析和依存句法分析的关系模式抽取关系元组的方法，克服了单纯依赖词法分析的关系抽取方法无法处理关系词与实体距离较远的问题。同时本文通过自动构建学习语料，学习得到大量高质量的关系模式。

(3) 本文提出了一个完整的中文开放式实体关系抽取算法 DPM。能够自动地从开放域中文语料中学习得到实体关系模式，并据此实现对语料中二元或多元实体关系的自动抽取。实验的结果表明，DPM 算法的性能优于当前主流中文开放式关系抽取算法。

(4) 本文实现了中文关系抽取系统，系统一方面利用百科网页抽取了大量的属性关系元组、上下位关系，另一方面从网络纯文本中抽取了大量的实体关系元组，并构建了一个关系知识库。

## 6.2 不足与下一步展望

本节将通过分析实验数据中错误结果，进而分析本文提出算法的不足之处以进一步提出改进之处。具体的本节将分析 4.3.1 节中 Logistic 回归阈值取 0.22 时，算法在 Wiki-500 语料上的输出结果。本文对错误结果进行了人工比对，得到如下错误类型：

第一类错误是实体短语和关系词短语不匹配，占错误元组总数的 66%。例如语句：“排除家长因为疾病抛弃孩子的可能性”，抽取结果为(疾病，抛弃，孩子)。

第二类错误是由于关系抽取时使用的模式选择错误或者没有模式与之匹，占错误元组总数的 5%。例如，语句“欧洲经济危机时期，波兰保持了经济竞争力，且通过举办欧锦赛向世界展示波兰”，由于没有与之相匹配的关系模式，导致(波兰，举办，欧锦赛)该关系元组没有抽取出来。

第三类错误是自然语言处理工具带来的错误，包含分词错误，词性标记错误和依存句法分析错误等等，占错误元组总数的 29%。例如句子“黎曼猜想是数学最基本的未决问题之一”，由于分词工具无法将“黎曼猜想”标记为名词，因而抽取结果为(黎曼，猜想，数学数学最基本的未决问题之一)。

错误分析结果表明，本文学习得到的模式准确率和覆盖率方面还有待提升。另一方面，为了提高模式的准确性，我们将进一步加入模式的筛选算法。不难看出，自然语言处理工具的性能是限制关系抽取算法性能的主要原因之一，对自然语言处理的深入研究也是改善方法性能的方向之一。此外进一步增加关系知识库中的关系元组，以及基于知识库的应用也是未来的研究方向。

# 致 谢

在这三年里，我收获颇多，对许多人给予我的诸多帮助，我一直心存感激。

首先，我要感谢耿技老师的知遇之恩，能师从耿技老师是我一生之福。耿技老师一次次对我的鼓励、开导以及潜移默化的影响，使得我相对以前更加成熟和自信。这一点在使得我无论在项目中还是实际生活中遇到问题，都可以静下心来去思考解决方案。我要衷心的感谢我的恩师刘峤老师，由于自己的任性和不成熟，曾多次令其失望，尽管如此，刘峤老师仍对我寄予厚望，悉心教导，感谢刘峤老师对我的包容。在学术研究和项目工作上，我还得到过刘瑶老师、周尔强老师以及秦臻老师帮助和建议，感谢老师们。

回首过去的三年，求学路上幸运的遇到了很多乐于帮助我的同学和朋友。感谢夏勇、罗熹、邓丽妮等师兄师姐，不厌其烦的在学术上和生活上对我的帮助；感谢同门鲍晨阳、刘庆、高红和康晓慧在方方面面的帮助和支持，陪伴我顺利地走完研究生的求学道路；感谢钟云、李佩伦、江浏祎等师弟师妹们的热情帮助和合作，特别是田晟兆小师弟在投论文期间和我一起在实验室熬了几个通宵；感谢信息安全班级的全体同学，感谢好友赵永福、杨韵硕、吴世坤等让我拥有如此美好的一段校园生活；还要特别感谢经常晚上帮我开主楼大门的大叔以及帮我开宿舍大门的宿管阿姨，同时在这里也和你们说声抱歉，经常夜里打扰到你们；感谢知识图谱领域的大拿王昊奋老师，经常在微博上为我解除自己研究领域的疑惑。

我要衷心感谢我的父母，感谢他们为我创造了良好的学习条件，感谢他们多年来的养育之恩和无私奉献。

感谢这三年的时光，感谢在电子科技大学遇到这么多可爱的人！

# 参考文献

[1] C. Bizer, T. Heath,T. Berners-Lee. Linked data-the story so far [J].Int Journal on Semantic Web and Information Systems, 2009,5(3):1-22

[2] 王元卓, 贾岩涛, 刘大伟, 等. 基于开放网络知识的信息检索与数据挖掘 [J]. 计算机研究与发展, 2014, 52(2): 456-474

[3] N. Chinchor, E. Marsh. Muc-7 information extraction task definition[C] Message Understanding Conference, FairFax, 1998: 359-367

[4] L. F. Rau. Extracting company names from text[C] Artificial Intelligence Applications Conference Proceedings,New York, 1991:29-32

[5] X. H. Liu, S. D. Zhang, F. R. Wei, et al. Recognizing named entities in tweets[C] Association for Computational Linguistics: Human Language Technologies,Portland,2011: 359-367

[6] Y. F. Lin, T. Tsai,W. C. Chou, et al. A maximum entropy approach to biomedical named entity recognition[C] SIGKDD Workshop on Data Mining in Bioinformatics, Seattle, 2004:56-61

[7] S. Sekine, K. Sudo, C. Nobata. Extended named entity hierarchy[C] Language Resources and Evaluation Conference, New York, 2002:1818-1824

[8] X. Ling,D.S Weld. Fine-grained entity recognition[C] Association for the Advancement of Artificial Intelligence, Toronto, 2012: 94-100

[9] 赵军, 刘康, 周光有,等.开放式文本信息抽取[J].中文信息学报, 2011, 25(6): 98-110

[10] C . Whitelaw, A . Kehlenbeck, N. Petrovic, et al. Web-scale named entity recognition[C] ACM Conference on Information and Knowledge Management,California, 2008:123-132

[11] N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations[C]Association for Computational Linguistics,Barcelona, 2004:1-22

[12] R. C. Bunescu, R. J. Mooney. A shortest path dependency kernel for relation extraction[C] Human Language Technology and Empirical Methods in Natural Language Processing, New York, 2005: 724-731

[13] Y. M. Zhang , J. F. Zhou. A trainable method for extracting Chinese entity names and their relations[C] Association for Computational Linguistics,Paris, 2000:66-72

[14] M. Banko, M. J. Cafarella,S. Soderland, et al. Open information extraction for the Web[C]Internatioal Joint Conf on Artificial Intelligence, Hyderabad, 2007:2670-2676

[15] F. Wu, D. S. Weld. Open information extraction using Wikipedia[C] Association for Computational Linguistics,Uppsala, 2010: 118-127

[16] A. Fader, S. Soderland, O. Etzioni. Identifying relations for open information extraction[C] Empirical Methods in Natural Language Processing,Edinburgh,2011:1535-1545

[17] M. S. Mausam, R. Bart,S. Soderland, et al. Open language learning for information extraction[C] Empirical Methods in Natural Language Processing and Computational Natural Language Learning,Jupe,2012: 523-534

[18] M. Banko,O. Etzioni. The Tradeoffs between open and traditional relation extraction[C] Association for Computational Linguistics,Columbus, 2008: 28-36

[19] J. Zhu, Z. J. Nie, X. J. Liu, et al. StatSnowball: A statistical approach to extracting entity relationships[C] World Wide Web,ADRID,2009:101-110

[20] Y. H. Tseng,L. H. Lee,S. Y. Lin, et al. Chinese Open Relation Extraction for Knowledge Acquisition [C]Association for Computational Linguistics European,Sweden,2014,12-16.

[21] L. K. Qiu,Y. Zhang. ZORE: A Syntax-based System for Chinese Open Relation Extraction[C] Conference on Empirical Methods in Natural Language Processing,Doha,2014,1870-1880.

[22] A. Alan, L. Alexander. KrakeN: N-ary facts in open information extraction[C] Automatic Knowledge Base Construction and Web-scale Knowledge Extraction,Jeju,2012:52-56

[23] A. McCallum. Joint inference for natural language processing[C] Conference on Computational Natural Language Learning,Colorado,2009:1-1

[24] 张培颖, 李村合. 基于知识库的交集型歧义字段切分系统[J].计算机系统应用, 2006(8):42-43.

[25] 何克抗, 徐辉, 孙波. 书面汉语自动分词专家系统设计原理[J].中文信息学报, 1991(2):1-14.

[26] 张茂元, 卢正鼎, 邹春燕. 一种基于语境的中文分词方法研究[J].小型微型计算机系统, 2005, 26(1):129-133.

[27] 王伟, 钟义信, 孙建,等. 一种基于EM非监督训II练的自组织分词歧义解决方案[J].中文信息学报, 2001, 15(2):38-44.

[28] H. P. Zhang , H. K. Yu ,D. Y. Xiong, et al. HHMM-based Chinese lexical analyzer ICTCLAS[C] Proceedings of the second SIGHAN workshop on Chinese language processing,Sapporo, 2003: 184-187.

[29] B. B. Greene,G. M. Rubin. Automated grammatical tagging of English[M]. Rhode Island:Brown University,1971

[30] E. Brill. Transformation-based error-driven learning and natural language processing: a case

study in part-of-speech tagging[J]. Computational Linguistics, 1995, 21(4):543-566.

[31] T. Moon,K. Erk,J. Baldridge. Crouching Dirichlet, Hidden Markov Model: Unsupervised POS Tagging with Context Local Tag Generation[C]Conference on Empirical Methods in Natural Language Processing,Cambridge,2010:196-206.

[32] 袁里驰. 基于改进的隐马尔科夫模型的词性标注方法[J].中南大学学报:自然科学版, 2012, 43(8):3053-3057.

[33] 姜尚仆,陈群秀．基于规则和统计的日语分词和词性标注的研究[J].中文信息学报, 2010, 24(1): 117-123.

[34] 张卫. 中文词性标注的研究与实现[D].南京：南京师范大学, 2007.4

[35] 李彬, 刘挺, 秦兵,等. 基于语义依存的汉语句子相似度计算[J]. 计算机应用研究, 2003, 20(12):15-17.

[36] W. Che, V. I. Spitkovsky, T. Liu. A comparison of Chinese parsers for Stanford dependencies[C] Association for Computational Linguistics,Jeju,2012: 11-16

[37] R. McDonal, K. Crammer, F. Pereira . Online large-margin training of dependencyparsers[C] Association for Computational Linguistics,Michigan,2005: 91-98

[38] R. McDonal,F. Pereira. Online Learning of Approximate Dependency Parsing Algorithms[C]. European Chapter of the Association for Computational Linguistics, Trento, 2006:81-88

[39] T. Koo, M. Collins. Efficient Third-Order Dependency Parsers[C]. Association for Computational Linguistics,Uppsala,2010,1–11

[40] X. Qiu,Q. Zhang,X. Huang. FudanNLP: A Toolkit for Chinese Natural Language Processing[C] Association for Computational Linguistics,Sofia,2013,49-54.

[41] T. Liu, W. X. Che, Z. H. Li. Language Technology Platform[C]Computational Linguistic Conference Proceedings,Beijing,2010,13-16.

[42] L. K. Qiu,Y. Zhang,P. Jin , et al. Multi-view Chinese Treebanking[C]Computational Linguistic Conference Proceedings,Dublin,2014,257-268.

[43] Z. Huang. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values[J]. Data Mining & Knowledge Discovery, 1998, 2(3): 283-304.

[44] T. Zhang,Birch.An efficient data clustering method for very large databases[C]ACM SIGMOD,Montreal,1996,103-114.

[45] M. Ester,H. P. Kriegel,J. Sander, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]Knowledge Discovery and Data Mining,Portland,1996, 96, 226-231.

[46] W. Wang,J. Yang,R. R. Muntz. STING: A Statistical Information Grid Approach to Spatial Data

Mining[C]Very Large Data Bases,Athens,1997,186-195.

[47] Jr. D. W. Hosmer, S. Lemeshow, R. X. Sturdivant. Applied logistic regression[M]. John Wiley & Sons, 2013

# 攻硕期间取得的研究成果

发表论文：

[1] 刘峤、李杨、刘瑶等，知识图谱构建技术综述[J]，计算机研究与发展，2016,53(3)：582-600.

[2] 刘峤、钟云、李杨等，基于图的中文集成实体链接[J]，计算机研究与发展 2016,53 (2):270-283

科研项目：

[1] 863 计划. 国家高技术研究发展计划主题项目. 2011AA010706. 2011

[2] 自然基金项目. 面向移动社会网络的多尺度交叉感知计算理论与关键技术. 61133016. 2011

申请专利：

[1] 《一种基于模式自学习的中文开放式关系抽取方法》 专利号：201510475450.0

[2] 《一种基于依存分析的中文兼类词处理方法》专利号：201510475708.7

[3] 《一种基于图模型的中文集成实体链接方法》专利号： 201510475469.5