

密级：公开

# 口语对话系统 EasyNav 的研究与实现

(申请清华大学工学博士学位论文)

培 养 单 位： 计算机科学与技术系

专 业： 计算机应用技术

研 究 生： 黄 寅 飞

指 导 教 师： 吴 文 虎 教授

副指导教师： 郑 方 副教授

二零零二年四月

# **The Research and Implementation of Spoken Dialogue System: EasyNav**

Dissertation Submitted to

**Tsinghua University**

in partial fulfillment of the requirements

for the degree of

**Doctor of Engineering**

**by**

**HUANG Yinfei**

**( Computer Application )**

Dissertation Supervisor: WU Wenhui, ZHENG Fang

## 中文摘要

本文研究并实现了一个口语对话系统 EasyNav，重点研究其语言理解部分。EasyNav 场景为清华校园内地点信息查询服务，具有真实应用背景，其文本特点是用户主导的连续查询问句。

本文贡献之一是提出互连框架方法，进行较高层面的语义分析与表示，能够对后续查询与应答处理提供指导，能够有效支持上下文分析。互连框架方法的特点是：1) 互连框架、查询链和主题结构依据意义组合原则提出，具有较强的描述能力；2) 语义表示单位具备意义计算能力，分析过程中能够动态生成整句意义计算信息；3) 双层多遍的语义分析流程与上下文分析紧密耦合，易于抽取上下文分析所需要的部分语义表示。互连框架方法能够分析多层或递归的复杂句子和包含省略现象的连续提问，对符合场景的输入分析正确率达到 96.76%。互连框架在表示能力上要强于其它系统常用的语义框架和格框架。

本文贡献之二是考察汉语省略现象特点后提出语境省略的概念，强调：1) 基于语义知识进行省略检测；2) 立足于当前句进行省略重构。提出省略发现、筛选、省略校验和倾向选择四步骤框图以概括语境省略分析的处理要素。本文对于汉语口语对话下省略现象的研究具有原创性。

本文贡献之三是提出主题结构方法，研究汉语口语对话背景下的省略分析方法。主题结构方法的特点是：1) 符合语境省略概念，适合处理汉语口语对话中出现的省略现象；2) 对主题变化及其对约束条件延续的作用进行建模；3) 通过“省略类型=>省略成分=>省略性质”的拆解，建立一般性的处理框架。主题结构方法能对场景内的各种省略类型及其组合进行一致的处理，对符合场景的输入分析正确率达到 91.85%。主题结构方法对主题变化建模问题的研究具有前瞻性。

本文还对设计实现口语对话系统 EasyNav 过程中的其它工作作了介绍，包括：对系统流程的整体设计、基于 GLR 的句法分析、对话管理、应用查询、应答生成。目前系统为文本输入输出条件下能工作的完整版本。

关键词：口语对话系统，语音理解，人机交互，语义表示，省略分析

## **Abstract**

In this paper, a spoken dialogue system EasyNav is studied and implemented. The research is primarily carried out on the language understanding part of the system. The EasyNav domain is campus navigation information querying, which is oriented to real applications. The style of the text in the domain is user-initiative continuous querying.

The first contribution of the paper is presenting the Interlinked Frames Method to solve the problem of semantic parsing and meaning representation in a high level, which could guide the behaviors of the following components of the application query and the response generation, and provide efficient supports on the context analysis. The characteristics of the Interlinked Frames Method are: 1) The Interlinked Frames, Query Chains and Theme Structure presented for the meaning representation all have strong describing abilities because they all obey the Meaning Combination Principle; 2) The meaning representation unit contains the meaning calculation information, which could be combined to generate the meaning calculation information of the whole sentence during the semantic parsing; 3) The bi-stage multi-cycle process of the semantic parsing is tightly integrated with the process of the context analysis, and the partial meaning representation which is crucial for the context analysis can be easily abstracted. The Interlinked Frames Method has the ability of processing complex sentences that have a deep hierarchical level or a recursive structure, and the ability of processing continuous queries that contain the ellipsis phenomena. The semantic parsing success rate over the in-domain inputs is 96.76%. The Interlinked Frames outperform the Semantic Frames and the Case Frame in their representation abilities.

The second contribution of the paper is presenting the concept of Contextual Ellipsis by studying the Chinese ellipsis phenomena. The concept of Contextual Ellipsis emphasizes: 1) doing the ellipsis detection based on the semantic knowledge; 2) doing the ellipsis reconstruction from the current sentence. A 4-step process of the ellipsis detection, the filter, the ellipsis validation and the preference, is presented to generalize the basic elements of the Contextual Ellipsis resolution. The research on the ellipsis phenomena of the Chinese spoken dialogue in this paper is original.

The third contribution of the paper is presenting the Theme Structure Method to solve the problem of ellipsis resolution in the Chinese spoken dialogue. The characteristics of the Theme Structure Method are: 1) The method is based on the concept of Contextual Ellipsis thus is suitable to deal with the ellipsis phenomena in the Chinese spoken dialogue; 2) The method models the theme variation and its effects on the continuation of the constraints; 3) A general ellipsis resolution framework is established through the decomposition from ellipsis types to ellipsis elements and further to ellipsis properties. The Theme Structure Method provides a uniform resolution on various ellipsis types and their combinations in the domain. The ellipsis resolution success rate over the in-domain inputs is 91.85%. The research contains several insights on the relationship between the theme variation and the ellipsis detection.

The paper also introduces other works on the design and the implementation of the spoken dialogue system EasyNav, including: the whole system design, the GLR syntactic parser, the dialogue management, the application query and the response generation. Currently the system works well under the condition of written input and output.

Key words: Spoken Dialogue System, Speech Understanding, Human-Computer Interaction, Meaning Representation, Ellipsis Resolution

# 目 录

第一章 引言 .....	1
1.1 口语对话系统的研究意义 .....	1
1.2 口语对话系统概貌 .....	1
1.2.1 基本语音语言技术 .....	1
1.2.2 系统开发技术 .....	3
1.2.3 面向应用技术 .....	3
1.3 国内外口语对话系统综述 .....	4
1.3.1 欧委会支持项目 .....	4
1.3.1.1 SUNDIAL 计划 .....	4
1.3.1.2 MASK 计划 .....	4
1.3.1.3 RAILTEL 与 ARISE 计划 .....	4
1.3.2 美国 DARPA 支持项目 .....	5
1.3.2.1 ATIS 计划 .....	5
1.3.2.2 Communicator 计划 .....	5
1.3.3 研究机构 .....	6
1.3.3.1 MIT-SLS .....	6
1.3.3.2 Rochester-CISD .....	6
1.3.3.3 LIMSI .....	6
1.3.3.4 KTH 和 NLPLAB .....	7
1.3.3.5 Lucent .....	7
1.3.3.6 AT&T .....	7
1.3.3.7 国内的研究单位 .....	7
1.4 本文研究的问题 .....	8
第二章 系统概况 .....	10
2.1 系统结构 .....	10
2.2 模块功能简介及数据流 .....	11
2.3 设计考虑 .....	13
2.3.1 单回合交互模式 .....	13
2.3.2 部分语言系统 .....	15
2.3.3 规则方法 .....	15

2.3.4 意义组合原则 .....	16
2.3.5 知识表示 .....	17
第三章 句法分析 .....	19
3.1 预处理 .....	19
3.2 句法规则 .....	19
3.3 GLR 分析器 .....	22
3.4 句法树和句法约束条件 .....	25
3.5 句法分析的作用 .....	26
第四章 语义分析与表示 .....	28
4.1 语义分析的四个层面和意义组合原则 .....	28
4.2 句型模板方法 .....	29
4.2.1 整句框架与句型模板 .....	29
4.2.2 句型模板规则 .....	30
4.2.3 模板匹配过程 .....	32
4.2.4 方法评价 .....	33
4.3 互连框架方法 .....	34
4.3.1 互连框架和主题结构 .....	34
4.3.1.1 互连框架 .....	34
4.3.1.2 查询链 .....	36
4.3.1.3 主题结构 .....	38
4.3.2 语义规则 .....	40
4.3.3 语义分析过程 .....	44
4.3.3.1 分析器 .....	45
4.3.3.2 分析过程 .....	47
4.3.3.3 与句型模板方法的比较 .....	50
4.3.4 方法评价 .....	50
4.4 其它语义分析表示方法 .....	50
4.4.1 MIT-SLS .....	51
4.4.2 LIMSI .....	52
4.5 小结 .....	52
第五章 省略分析 .....	53
5.1 引言 .....	53
5.1.1 指代 .....	53

5.1.1.1 指代现象 .....	53
5.1.1.2 指代分析 .....	54
5.1.2 省略 .....	55
5.1.2.1 省略现象 .....	55
5.1.2.2 省略分析 .....	56
5.1.3 会话理论 .....	57
5.1.3.1 会话 .....	57
5.1.3.2 会话结构 .....	58
5.1.3.3 会话关系 .....	58
5.1.3.4 中心理论 .....	59
5.2 语境省略 .....	60
5.2.1 汉语和英语中省略的不同 .....	60
5.2.2 语境省略定义 .....	62
5.3 省略分析 .....	64
5.3.1 对于主题结构的讨论 .....	64
5.3.1.1 主题变化 .....	64
5.3.1.2 关注状态 .....	65
5.3.2 省略分析处理步骤 .....	66
5.3.3 基于主题结构的分析 .....	67
5.3.3.1 省略发现 .....	67
5.3.3.2 筛选 .....	68
5.3.3.3 省略校验 .....	68
5.3.3.4 倾向选择 .....	69
5.3.3.5 疑问词省略 .....	70
5.3.3.6 寻找引用项失败的处理 .....	70
5.3.3.7 重构过程 .....	70
5.3.3.8 更新过程 .....	71
5.3.3.9 细流程 .....	72
5.3.4 统一的处理框架 .....	73
5.3.4.1 多约束条件查询 .....	73
5.3.4.2 缺省值 .....	75
5.3.4.3 片断 .....	75
5.3.4.4 零代词 .....	76

5.3.4.5 从到句 .....	76
5.3.4.6 动词及其施事受事 .....	78
5.3.5 几点补充 .....	79
5.3.5.1 指代 .....	79
5.3.5.2 现象的不同定义 .....	79
5.3.5.3 文体与省略现象 .....	80
5.4 小结 .....	80
第六章 对话管理与查询应答 .....	81
6.1 对话管理 .....	81
6.2 地图知识数据库设计 .....	81
6.3 第一版中的查询应答 .....	83
6.3.1 查询 .....	83
6.3.2 应答生成 .....	84
6.4 第二版中的查询应答 .....	84
6.4.1 动态查询组织 .....	85
6.4.2 应答中的错误处理 .....	86
第七章 系统评估 .....	88
7.1 目的与方法 .....	88
7.2 语料 .....	88
7.3 评估 .....	89
7.3.1 交互成功的统计 .....	89
7.3.2 交互不成功的统计 .....	90
7.3.3 交互不成功原因统计 .....	90
7.3.4 语义分析成功率 .....	92
7.3.5 上下文分析成功率 .....	92
第八章 结论 .....	94
参考文献 .....	96
致 谢 .....	104
个人简历 .....	105

## 第一章 引言

### 1.1 口语对话系统的研究意义

随着互联网的飞速普及，人们希望能在任何时间、任何地点、用本国语言方便地获取信息。

理想的解决方案就是提供一个具有听说能力的小型便携设备，该设备能通过和人的交谈提供所需要的信息。这种设备的核心技术是口语对话系统。口语因其自然、有效、灵活、便利而成为最合适的交互方式。[1]

口语对话系统是基于语音的人机接口，涉及的技术不仅仅是语音识别和语音合成，还包括自然语言处理和人机交互接口技术。相关技术的趋向成熟推动了口语对话系统的研究。特别是大词汇量、非特定人、连续语音识别技术的成熟，推动了语音技术实用化的研究方向。而对口语对话系统的研究，反过来也会在真实应用的背景下对于各项技术提出新的要求。尤其是语音、语言处理技术的结合，对于两个方向的研究都提出更多的挑战。

近十年来，口语对话系统的研究成为一个热点，很多家研究机构投入其中，推出的系统多达上百个。同时还有多家研究机构共同参与的联合计划，通过定期评测推动系统性能的稳步提高。ICSLP、EUROSPEECH 等国际会议上每年有上百篇相关文章，并有许多专题性的小型会议如 ISSD、ESCA 工作组，各种语音学术杂志上也经常推出口语对话系统主题的专辑。

### 1.2 口语对话系统概貌

口语对话系统涉及多门学科，多项技术，以下分别从基本语音语言处理技术、系统开发技术和面向应用技术三个角度进行介绍。

#### 1.2.1 基本语音语言技术

与口语对话系统相关的技术包括：语音识别、语言理解、对话管理、语言

生成、语音合成等。其中语言理解部分又分为句子理解部分和上下文理解部分。下面重点介绍口语理解相关的各项技术，考察技术的现状，以及口语对话系统对该项技术的要求。

语音识别的研究包括声学特征提取、声学建模、语言建模、语音搜索等方面。MelCep 作为特征、Tri-Phone 作为识别基元进行上下文相关建模、HMM 作为声学模型、N-Gram 作为语言模型，是目前语音识别技术的主流。HTK 的推广使用，也使得更多研究机构能够在一个比较高的起点上进行语音识别技术的研究。

口语对话系统中的句子理解有以句法为主和以语义为主两种方法。前者使用 CFG 得到整句结构，如 TINA[2]。后者考虑到语音输入可能会不可靠，而采取抽取关键词或关键短语再进行组合的方式，如格框架语法[3]和部分分析。上下文理解则要解决连续对话过程中的指代省略问题和信息延续问题。

语音识别和语言理解间的接口通常为首选句子、N-best 多选或者词网格。对于 N-best 多选，可以以符合句法作为条件进行筛选。概率文法也常被使用，以便在语音解码过程中提早用到语言知识。

在口语对话系统背景下，对语音识别与理解技术有更高的要求。首先是要能处理不流利现象，如重复、修补、解释、重说、嗯啊等，有些可以在语音层上建立垃圾模型解决，有些可以在语言层上根据语言知识检测加以解决。其次，要考虑说话人与信道变化，使用鲁棒性或自适应方法解决。

此外，人-人对话中的一些现象，也希望人-机对话中能够给予支持以提高系统的自然程度。一个是打断 (Barge-in)，即允许用户在系统说话的时候打断，这需要回波消除技术以及相应的对话策略支持[4]。一个是后台应答 (Back-channel)，即在系统说话时用户可以用简短的插话表示自己已经听到。

对话管理规划系统和用户间的交互过程。系统主导指完全由系统来控制交互过程，常见于早期的菜单式电话系统。用户主导指系统处于被动地位，只是对用户的请求进行回答。目前的主流是混合主导，即系统和用户都可以在一定程度上主导交互过程。自适应主导也属于混合主导，系统可以考察对话进程切换主导策略[5]。

口语对话系统中的语言生成通常并不复杂，采用模板生成方法即可满足要求。语音合成，目前比较好的方式是从语料中根据某种扭曲量度抽取合适的基元进行拼接，用这种方式合成的语音比用算法生成的语音更自然。

### 1.2.2 系统开发技术

语料收集对于系统的开发、训练和评估有重要作用，但在系统开发成功以前，无法得到真实的人-机对话语料。解决的办法一是使用人-人对话语料，但由于人-人对话和人-机对话有很大差别[6]，这一方法只能作为一个参考；另一是使用模拟人-机对话语料(Wizard of Oz, WOZ)，即让人代替计算机去与用户对话，并告诉用户这是在和计算机对话[7]，WOZ 语料要比人-人对话语料更接近人-机对话语料。在初步建立系统之后，就可以收集到真实的人-机对话语料，并针对这些语料来改进系统，从而形成一个逐步完善的循环过程。

评估，一方面是对系统开发的进展进行考察，另一方面通过公共的量度使得对不同系统性能的比较成为可能。评估量度主要分客观量度和主观量度两类。客观量度包括对整个系统性能的量度，如对话成功率、回合数、修改次数、句子长度等，以及各个模块性能的量度，如词错误率、槽错误率等。主观量度通过用户填写的问卷调查表获得，用于考察用户对系统的满意程度。PARADISE 评估框架试图将主观量度和系统性能的客观量度统一起来进行评估[8]。

### 1.2.3 面向应用技术

近年来口语对话系统在教育角度的研究增多，重点放在如何使系统更好用、更容易开发上。多模式输入输出技术是要为用户提供更方便友好的界面。分布式系统架构则为普遍式的移动系统提供了支持。对人-人对话模型的研究希望能使得系统行为更人性化。开发工具的研究则希望能突破口语对话系统知识密集的开发模式，能为特定应用快速开发出系统来。

近年来随着口语对话系统的发展，多模式输入输出成为一个研究热点，ICSLP'2000 的主题即为多模式对话系统。多模式，指不仅是语音输入输出，还考虑手势、指示、书写、键入、触摸、感知、摄像等多种输入，文字、图形、人脸等多种输出。此时，要考虑语音输入和其它输入间的组合关系。

口语对话系统目前技术水平只能针对特定领域特定任务，开发工作量很大，而且需要大量语言技术方面的专业知识，因此用开发工具使领域专家快速搭建系统便成为很有价值的研究方向。比较有名的开发工具有：OGI 的 CSLU 工具包[9]、MIT 的 SpeechBuilder[10]、LOTOS[11]等。

### 1.3 国内外口语对话系统综述

下面介绍有影响力、有代表性的口语对话系统。先介绍两个欧洲和美国的主要联合研究计划，然后介绍一些著名的研究机构。

#### 1.3.1 欧委会支持项目

自 1984 年以来，欧委会在研发方面支持了一系列框架计划，每个持续 4-5 年：FP1 (1984-1987) \ FP2 (1987-1991) \ FP3 (1990-1994) \ FP4 (1994-1998) \ FP5 (1998-2002)。框架计划中与口语处理和人机通讯相关的主要有两个研究计划：ESPRIT 研究计划，后改称为 IT 研究计划；TELEMATICS 研究计划中的 LR 研究计划[12]。下面我们具体介绍其中口语对话系统方面的一些计划。

##### 1.3.1.1 SUNDIAL 计划

SUNDIAL (Speech UNderstanding in DIALogue) 是早期的口语对话系统研究计划 (1988-1993) [13]。电话系统，领域为航班预订、航班查询和火车查询。支持法、德、意、英四种语言。系统模块完备，对语料收集和系统评估也作了研究。

##### 1.3.1.2 MASK 计划

MASK (Multimodal-Multimedia Automated Service Kiosk) 计划 (1994-1997)，是服务台系统，领域为火车查询订票[14]。系统试图创造一种新型的多模式输入、多媒体输出的公共服务设施形式。系统被安置在巴黎的 Gare St. Lazare 街道进行实际测试[12]。

##### 1.3.1.3 RAILTEL 与 ARISE 计划

RAILTEL (Railway Telephone Information Service) 计划[15]，以及其后续的 ARISE (Automatic Railway Information Systems for Europe) 计划 (1996-1998) [16]，是电话系统，领域为火车查询订票。ARISE 计划以三种语言实现了多个系统，其中包括分别由 LIMSI 和 Philips 公司开发的两个法语系统，Philips 公司开

发的荷兰语系统 VIOS，CSELT 开发的意大利语系统。

通过多家研究机构的参与，以及各自不同的设计策略、大规模真实用户的测试评估，ARISE 计划对口语对话系统设计的很多方面进行了探索，对用户行为和系统策略间的互动有相当的认识[17][18]。

### 1.3.2 美国 DARPA 支持项目

#### 1.3.2.1 ATIS 计划

DARPA-SLS (Spoken Language System) 计划 (1989-1995) 是早期的口语对话系统，领域为航空旅行信息服务，计划偏重语音识别和理解技术的研究[19]。参与机构有 BBN、CMU、MIT、SRT、ATT、MITRE 和 Unisys。

#### 1.3.2.2 Communicator 计划

Communicator 计划在 SLS 计划之后启动，目标是为战场上的士兵提供一种自然有效且方便易用的信息通讯方式[20]。技术目标为：有效的交互策略，要求交互自然且成功率高；知识引导的交互策略，要求能根据知识源主导对话。提出的关键技术难点包括：对话管理、上下文跟踪、语言生成、输入语言理解、免手和免眼交互。

初步研究的领域采用航空旅行规划和机票旅馆汽车预订。参与计划的研究机构接近 30 家，各自研究一部分技术，有 MITRE、AT&T、MIT、CMU、Lucent、IBM 等。其中 MIT 提供共享系统架构 Galaxy，这是一个分布式结构，以 Hub 为中心，提供具体功能的服务器与 Hub 相连，用脚本定义控制流程和服务器间信息传递的路线[21]。

计划每年召集参与机构统一进行系统评估与比较[22]。

### 1.3.3 研究机构

#### 1.3.3.1 MIT-SLS

美国 MIT 大学口语系统小组 (MIT-SLS) 很早就开始了口语对话系统方向的研究。在 1989 年推出的 Voyager 系统[23]提供 Boston 地区交通和导航信息查询服务, 标志了口语对话系统研究的开端。MIT-SLS 其后又推出了一系列面向不同领域任务应用的系统: Jupiter[24], 天气预报; Pegasus[25], 航班时间、入口信息查询; Mercury[26], 航班、票价信息查询; Orion[27], 离线自动通知系统; Wheels[28], 汽车广告检索; Dinex[29], 餐馆查询; WebGalaxy[30], 提供多领域信息查询的网络浏览器。

MIT-SLS 研究方向广泛, 语音识别、语言理解、对话管理、语言生成、语音合成等方向上都有成熟的技术。语言理解器 TINA[2], 系统架构 Galaxy[21]都具有很高的知名度。

#### 1.3.3.2 Rochester-CISD

美国 Rochester 大学交互与口语对话小组 (Rochester-CISD) 对口语对话的研究, 以后勤运输为背景, 着重研究机器辅助进行问题解决的能力。TRAINS 系统[31] (1991-1996) 领域为列车货运规划, 由于采用了对话动作分析和规划推理技术, 系统能够进行深层理解, 并混合主导对话。TRIPS 系统[32]作为其后续研究, 提供口语和图形界面交互, 扩展了领域的复杂度, 交通工具更多, 有时间、资源上的限制, 突出了用户与系统对话对于有效解决问题的必要性。

#### 1.3.3.3 LIMSI

法国 LIMSI 在口语对话系统方面的研究以信息获取为背景, 主要依托欧委会支持计划, 即前述的 MASK[14]、RAILTEL[15]、ARISE 计划[16]。系统主要采用基于格框架[3]的理解策略, 考虑缺省、简略回答、查询条件继承等上下文现象, 采用对话语法实现混合主导, 并有丰富的对话策略考虑。

### 1.3.3.4 KTH 和 NLPLAB

瑞典的 KTH (Royal Institute of Technology, 皇家技术学院) 提供语音和多媒体方面的技术, Linköping 大学的 NLPLAB 提供人机交互方面的技术。两个研究机构在口语对话系统方面有着联合的研究计划: 瑞典对话系统。

Waxholm[33]是 KTH 在 95 年完成的航船查询系统, 所采用的分析器 STINA 仿照 MIT-SLS 的 TINA 分析器设计。GULAN 是瑞典对话系统计划的一部分, 是由学生扩展对话系统的教学系统[34]。KTH 还开发了一系列人脸说话动作和语音同步输出的多模式口语对话系统, 有 AUGUST[35]、AdApt[36]等。NLPLAB 为口语对话系统提供自然语言对话框架体系 LINLIN[37], 以及支持多模式的框架体系 MALIN[38]。

### 1.3.3.5 Lucent

Lucent 贝尔实验室口语对话系统小组的研究偏重实用系统, 范围广泛, 有简单的电话路由系统[39], 也有比较复杂的信息查询、导航、交易系统, 还有提供个人信息消息服务的语音界面。

### 1.3.3.6 AT&T

AT&T 推出的自动接线系统[40], 以“ How may I help you? ”的无限制提示, 向非熟练用户提供服务。

### 1.3.3.7 国内的研究单位

中科院自动化所的旅游信息查询系统 Lodestar[41]是国内较早的口语对话系统, 有良好的演示效果。清华大学[42][43]、中科院声学所[44]、北方交通大学[45]等研究机构近年来也开展了口语对话系统方向的研究。

对口语对话系统方面的综述文献可以参考[46][47][1][48], [49][50]等书中也有关于对话的章节。

## 1.4 本文研究的问题

本文的论文工作主要是研究并实现一个口语对话系统 EasyNav。选择类似 Voyager 系统的校园地点信息查询应用作为场景。由于有真实应用为背景，以是否真正可用作为标准，测试、评估和进一步的完善都有明确的方向性。

相比只有单纯地点、路径查询的 Voyager 场景，EasyNav 场景中还包含了办公、服务查询，是一个多主题场景，这有点类似同时包含航空旅行规划、机票预订、旅馆预订和汽车预订的 Communicator 场景。由于对话过程中会出现频繁的主题变化，对此进行的研究也要更多地去考虑上下文问题。相比诸多的单主题表单式应用场景，EasyNav 系统所研究的问题具有前瞻性。

一个完整的口语对话系统通常由语音识别、语言理解、对话管理、语言生成、语音合成五个模块组成。由于组内语音处理有二十年的历史积累，而语言处理部分基本上没有基础，因此论文工作将重点放在口语对话系统中的语言处理问题上。

首先研究的语言处理问题是 parsing，即句法分析。句法分析提取句子的结构信息，是进一步理解的基础。句法分析本身是个比较大、比较困难的课题。我们没有把句法分析作为研究重点，一方面是因为 EasyNav 系统目前对结构信息并没有精细的要求，另一方面是因为语音识别模块尚未集成进系统中，句法分析研究缺乏对语音应用的针对性。

在口语对话系统研究实现过程中，我们发现语义分析和表示是语言理解的中心问题。语义表示是系统内部数据流中唯一计算机能理解并进行意义计算的形式。EasyNav 相对复杂的理解任务要求进行较高层面的语义分析，依据意义组合原则生成便于意义计算、包容上下文信息的语义表示。本文提出的互连框架方法达到了这一要求。

多主题场景要求系统有灵活的上下文处理能力，特别是主题变化时进行的局部上下文分析，即指代省略分析。汉语口语对话本身的特点使得我们主要关注语境省略现象。省略分析的难点在于需要确定是否存在省略，这需要综合句法、语义、上下文各方面的知识。另外省略类型及其组合的复杂度要求提出一个统一的处理框架。本文提出的主题结构方法达到了这些要求。

口语对话系统和自然语言查询接口的主要不同在于前者具有对话性，表现

为：1) 能依据自身具备的知识在适当的时候主导对话进程；2) 能在恰当的上下文中进行理解；3) 能作出合作性的应答。对话性由对话模块提供，将很大地影响人对于系统智能的感受。受时间精力所限，本文在这方面只进行了初步的考虑，做了部分的实现。

论文工作偏重基本技术角度的研究。

## 第二章 系统概况

### 2.1 系统结构

校园导航系统 EasyNav[42],是以提供清华校园内地点信息查询服务为背景的口语对话系统,可以对有关校园地点、办公、服务、路线等信息进行查询,不限定询问句型。EasyNav 系统目前发展到第二版。

第一版(1999.6–2000.3)的系统流程如图 2-1 所示。第一版以文本输入输出的对话系统为目标,采用通常汉语自然语言理解所用的流程[51]:分词、词性标注、句法分析、语义分析。

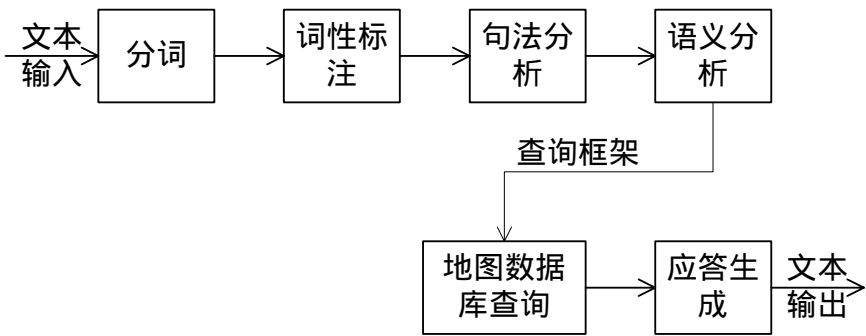


图 2-1：第一版的系统流程图

第二版(2000.3–2001.4)的系统流程如图 2-2 所示。流程中加入了语音输入输出、对话管理、上下文分析等模块。不过当前系统并未实现设计流程中的语音输入输出,对话管理模块功能也还比较弱。

在后续的各章中,会对句法分析、语义分析、局部上下文分析、查询应答等模块进行介绍。并未加入系统的语音识别和合成不介绍。对话管理目前主要是进行上下文管理,合在查询应答一章中介绍。分词、词性标注由于在系统中起的作用不大,合在句法分析一章中介绍。

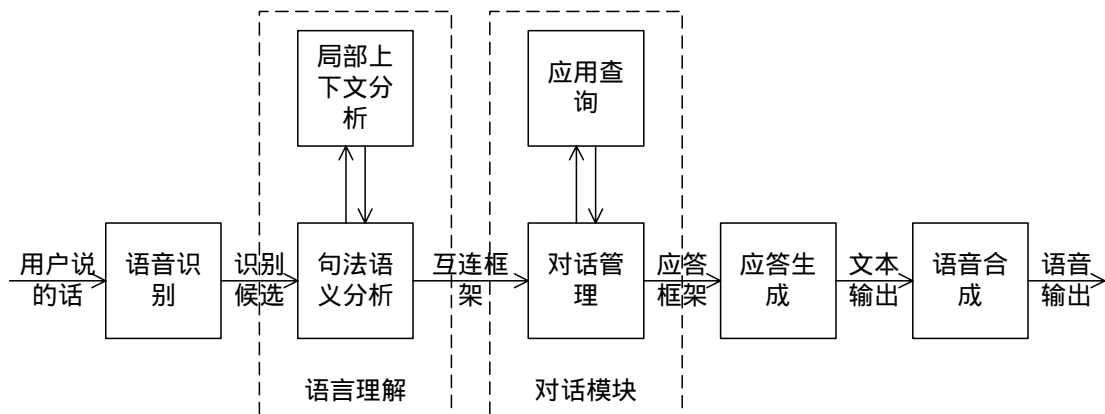


图 2-2：第二版的系统流程图

2.2 模块功能简介及数据流

图 2-3 在图 2-2 基础上给出了系统流程详图 ,图中列出了各个模块及与各模块相关的规则与数据流。

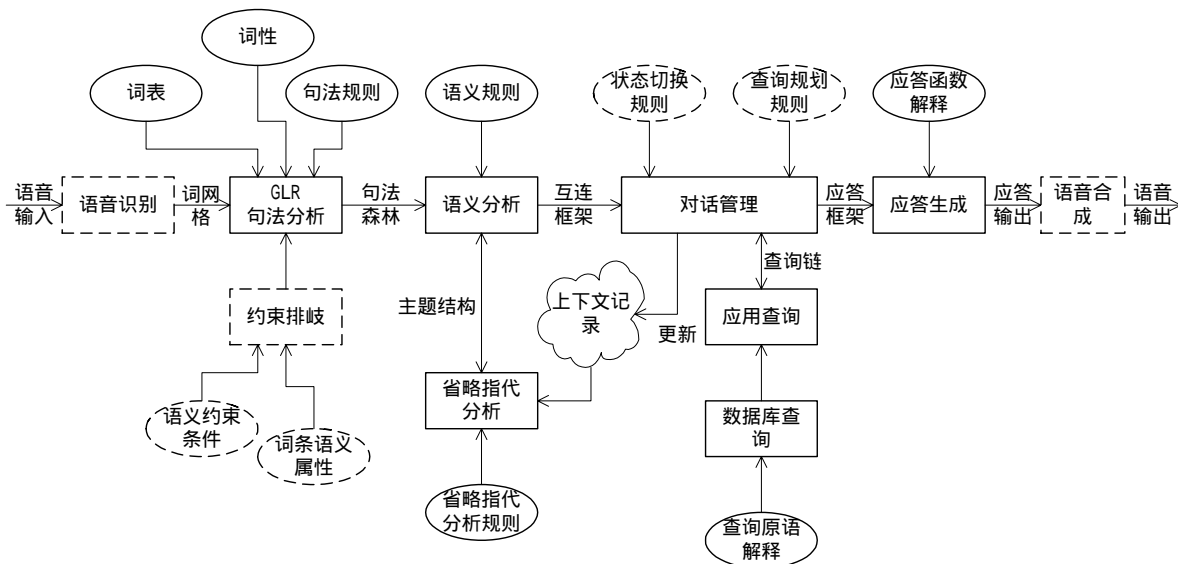


图 2-3：系统流程详图，虚线为尚未实现的部分

用户说的话通过采音设备转化成为语音信号输入。

**语音识别模块**将语音输入转化成为多条词序列候选，或是更为紧缩的词网  
格。

**句法分析模块**分析输入句子抽取结构信息。句法分析采用 GLR 分析器。利用 GLR 分析器的特性可以在语音识别的同时进行句法分析，即并行语音识别。句法分析模块所利用的知识除了句法规则之外，还有分词、词性标注预处理所需要的词表和词性知识，以及用于搜索过程中约束排歧的知识。约束排歧知识一方面包括一般性的词汇语义知识，如电子词典或者知网中的词条属性，另一方面包括应用场景所给予的语义约束条件，如词间搭配和固定用法。GLR 分析器的输出是若干句法分析树，压缩成为句法分析森林。

**语义分析模块**将输入句子中的语义成分依据语义规则进行组合[52]，建立对句子的语义表示——互连框架[53]。语义规则形式类似句法分析所用的 CFG，不过没有句法规则中要求的词序和邻接限制，但在部分规则中使用结构信息作为句法约束条件。互连框架是通过槽间连接相联系的一组框架，其内包含了语义信息及对语义信息的计算方法。语义分析过程也就是框架生成过程，采用双层多遍的流程，在 A 层进行本句语义成分的组合，在 B 层进行与上下文语义信息的组合。对于之前语音识别和句法分析产生的多个候选，以能在语义分析过程中从本句和上下文中抽取出最多的语义成分为原则，进行消歧。

**省略指代分析模块**在语义分析的 B 层循环中被调用，处理对话中的省略和指代现象，本文偏重省略分析[54][55]。该模块也被称为局部上下文分析模块(如图 2-2 中所示)，因其功能是将局部上下文中的语义信息与当前句中的语义信息进行组合。对于单回合交互模式而言，主题变化频繁，对上下文分析有很大影响。为此提出主题结构[53]，用以描述当前主题中的语义对象、属性和关系，并在此基础上研究主题变化的条件和对上下文引用的影响。为符合意义组合原则，省略指代分析规则附着在语义规则之中。

**对话管理模块**根据语言理解的结果确定系统的行为。状态切换规则用以确定应对用户言语行为的系统言语行为，常见的应对是“请求”对“回答”。为实现言语行为，通常需要调用应用查询模块完成用户的查询请求，并填充应答框架传递给应答生成模块生成应答。查询选择规则用以在查询失败的情形下重新规划查询以得到可以接受的查询结果。此外，对话管理模块还负责更新上下文记录，上下文记录将在下一回合中被省略指代分析模块使用。

**应用查询模块**根据互连框架中所包含的指示如何进行查询的查询链[53]实现查询操作[56][57]。查询链的组成单元是查询原语。查询原语附着于框架槽之上，语义分析槽间互连的同时也生成了查询链。查询操作通过对查询原语的解释来实现。部分查询原语的解释需要进行应用推理，如寻径、寻最近地点等。

**应答生成模块**根据互连框架中的应答框架组织应答[56][57]。应答框架中的应答入口通常指向一个应答函数，函数的参数指向查询链，查询完成之后参数都被填充，通过对应答函数的解释生成应答。对于查询失败后重新规划查询的情况，也有对应的应答模板，以生成智能的应答。

**语音合成模块**将生成的应答转化成语音输出。

整个流程是从左向右的单向流程，数据流以“语音->文本->语义表示->文本->语音”形式变化。图中虚线框的模块和规则尚未实现或尚未集成进系统中。流程主干能走通，因此将虚线部分集成进来对系统而言只是一个扩展完善的问题。

语义分析模块是理解的中心。语义规则中附着了句法约束条件、省略指代规则、查询原语、应答函数，这样在语义分析进行过程中，也就进行了上下文分析，同时生成了指导查询的查询链和指导应答的应答框架，后续的查询和应答模块所需要做的只是解释语义表示中所包含的相应信息。

对话管理模块是流程控制的中心。对话管理模块之前是单纯的理解、之后是单纯的应答，而具体系统该怎么做、该怎么表现出智能，都是对话管理模块的责任。状态切换规则决定系统采用何种言语行为，并能保留一定的全局上下文，从而允许系统进行主导。查询规划规则使得系统对用户输入信息不确切的情况具有鲁棒性，以提供一个正面的富含信息的对用户有帮助的应答（合作性应答）。此外上下文管理也由对话管理模块负责。

## 2.3 设计考虑

### 2.3.1 单回合交互模式

EasyNav 场景采用的是单回合交互模式，其文本的特点是连续的单回合查询问句。用户通常把要查询的内容及各种约束条件都在一个回合中给出，系统

完全可以使用这些信息进行一个完整的查询操作，并给予一个充分的应答。这种充分的应答指示用户查询目的已经达到，应答之后用户继续进行的查询激发一个新的查询目的，两个查询目的彼此无关。然而作为一个连贯的查询文本，前后两个查询间还可能存在语义信息上的关联，前一查询中提到的语义信息，在用户保留有一定印象的情况下，后一查询中可以省略。简而言之，前后两个查询间意图独立，但语义相关。

与此对立的是多回合交互模式，订票场景便是常见的一种。以航班查询为例，在系统主动询问或给予提示的情况下，用户一个回合提供一个或多个查询条件。当系统判断查询内容以及各种约束条件都齐全以后，进行航班查询。这一交互过程的目的是进行特定的航班查询，各句的目的是为查询提供查询条件，各句的目的是整个交互目的的子目的，因此在同一交互之内，前后两句间意图相关。在多回合交互模式下，以整个交互为单位进行语义表示。

在对话过程中主题的切换是常见的现象。我们定义主题是一个交互过程的用户意图，则主题的切换就是新的交互过程的开始，对应一个新的用户目的。以航班查询为例，举一个主题切换的例子，在订票过程中，有些事情不确定因而需要查询一下，则由订票主题切换到查询主题，中间又问了天气情况，切换到天气主题又切换回来，最后切换回订票主题。我们认为搜集参数时主题不变，触发实际的订票或查询操作后，主题才结束。比如说航班查询中，订了几张票后，又给定另外一组参数订了几张，则在第一次订票操作完成后，主题结束，下面的订票已经是新的主题了。这也就是同一主题类型，但不同主题的情况。因为单回合交互模式下一个回合就是一次交互，主题切换频繁，有必要对此给予特别的重视。

需要指出的，是单回合交互模式和多回合交互模式下语义和上下文处理的不同。在语义表示方面，单回合交互时，一个回合进行一次语义表示，而多回合交互时，需要将多个回合组成的交互作为一个整体进行语义表示。在上下文处理方面，单回合交互时，需要刻画前后两句间的语义相关性，多回合交互时，在一次交互内部因共有同一意图而对应有交互层的语义表示。以航班查询为例，多回合交互的最后一回合进行查询操作，所省略的查询参数是之前的回合提供并存入语义表示中的，省略信息已经在语义表示之中，不需要从上下文的语义表示中再抽取。

单回合还是多回合交互，取决于对话双方的行为。单回合交互模式通常在用户主导情况下使用，而这受系统行为影响，表现为系统不主动说第一句话，

或是问“您想问什么？”。多回合交互模式通常在系统主导情况下使用，系统行为表现为主动询问或提示，引导用户逐步提供信息。如果用户是新手，系统主导的多回合交互利于双方成功的对话；如果用户是老手，则对话会偏重于单回合的用户主导。

### 2.3.2 部分语言系统

系统的基本特点是面向任务的理解、领域受限，因而决定了系统是一个部分语言系统，这限定了系统各模块的功能设计目标。

有点类似蒙塔古语法[58]的研究方法，我们所研究的是部分语言系统。对自然语言系统整体的搭建，超出了当前技术能力，因此把自然语言具有代表性的子集作为研究对象，希望从这一子集中发现规律，进一步刻画整个自然语言语法语义体系。这一子集不是人工限定的，而是由场景天然限定的，因而其语言特点具有口语的典型意义，对其中复杂现象的进一步研究也具备从特殊到一般的可推广性。

具体到 EasyNav，我们就只研究 EasyNav 中通常出现的自然语言是如何组织实现的，主要研究背景是领域专家对系统功能的认知和实际搜集的语料。部分语言系统只是任务的简化而不是功能的简化，系统中依然包含了大量语言现象，需要有针对性地发现问题，解决问题。在 EasyNav 任务背景下解决的语言问题，如省略分析办法，具有推广到更广阔领域的的能力。为描述系统内部语义表示而提出的互连框架、主题结构、应答查询链等表示方法，具备对一般化语义的表示能力。

### 2.3.3 规则方法

在语言理解中通常倾向于使用规则方法而不使用统计方法[59]。基于规则的理解其优势在于能处理嵌套或多层的语义表示，能从带修补、重说等口语现象的“病句”中抽取出正确的语义，能处理指代、省略等基于上下文的推理。因此系统除了在分词和词性标注的预处理部分使用了统计方法外，整个理解过程都是使用规则方法，并且在进一步的完善中也将以规则方法为主进行设计。

为符合人的理解过程，我们保留句法分析，一是用其分辨出句子结构，帮助分析复杂句子；二是利用句法合法性作为语音解码中重要的剪枝条件。语义

分析不是对整个句法树作处理，而是根据自身需求抽取句法树中比较可靠的成分与关系。这样做，能够避免不可靠信息带来的错误，还能够利用场景知识加强对所需要信息的描述。在有限场景下，比较容易判断出用户的意图，因而获取用于查询的足够信息就可以认为达到了理解的目的。

*EasyNav* 中的语言理解流程设计要考虑句法分析和语义分析各自对理解作多大贡献，以及相互间如何关联。在流程中，句法分析利用基于词性的语法知识，进行松弛的分析，得到一个句法树。语义分析将关键词与句型模板匹配，在匹配过程中利用从句法树中抽取出来的句法约束条件来保证句子合法。

以目前的技术水平，还难以做到一般性的语音理解，因而口语对话系统通常面向一个有限场景。为使其实用化，快速搭建、易于扩展和移植也成为口语对话系统研究的重要目标。在本文的理解流程设计中，1) 使用一般性的句法规则；2) 句法分析和语义分析的规则与分析模块分离；3) 句法分析和语义分析相对独立。这都有利于上述目标的实现。

### 2.3.4 意义组合原则

意义组合原则[60]最早由弗雷格提出：句子的整体意义是它的部分意义以及它们组成方式的函数。这一原则认为意义是有结构的，不可穷举的。这也是我们在 *EasyNav* 系统中对意义进行分析和表示时的一个指导原则。

基于意义组合原则，系统中整句的语义表示是由各个语义成分自底向上、逐步组合的结果。考虑到上下文，不但要对本句中的语义成分进行组合，还要将本句语义成分与上下文中的语义成分进行组合。

我们认为问句的意义是查询义，即指示所需信息的查询表达式，查询义也要遵循意义组合原则。每个语义对象中都要定义如何对所包含的语义信息进行计算的方法，也就是每个对象的查询义。在语义对象的组合过程中，相应的计算方法也链接在了一起，当组合得到整句对应的意图对象时，整句的查询义也就通过这一链接过程得到了。

在 *EasyNav* 系统中，语义对象由互连框架表示，对象的查询义由框架内含的查询链表示，语义对象的组合由框架间的互连过程实现。随着框架互连，查询义也链接到了一起。整句的查询义由意图框架的应答入口所指示的应答查询链得到。

意义组合原则要求语义对象的表示具有自足性。对象要定义所包含信息的计算方法，还要保证在组合过程中计算方法能够正确地链接到一起。由于有可能是对上下文信息与本句信息进行组合，计算方法的链接过程要超越描述句内对象组合的语义规则，而更多地依赖于对象本身的完备表示。

### 2.3.5 知识表示

对话的过程就是对话双方以自身知识为基准去了解对方知识的过程，是一个知识共享的过程。图 2-4 是[48]中给出的对话模型图示，图 2-4 表明，对话过程实际上就是用户知识和系统知识的匹配过程。

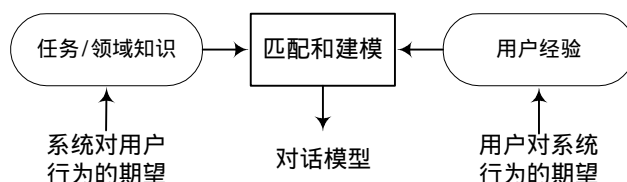


图 2-4：面向目标的对话建模

在这一过程中，由于双方知识的不一致，会给对话带来问题。用户可能对系统的功能局限不了解，而询问超出场景的范围，也可能对系统的任务细节不了解，而提出不恰当或信息不足的问题。对这些问题有两种处理方式，一是系统通过引导或者提示，将系统知识反馈给用户，一是系统发现这种不一致，并如实告诉用户。我们在 EasyNav 系统采用的是后一方法。发现不一致并将其反馈给用户的过程，我们称之为错误处理过程。

系统知识包含两层，一层是语言知识，一层是任务知识。具体到 EasyNav 系统，语言知识指的是词表和对每个词的语义解释，任务知识指的是地图知识数据库。语言知识比任务知识更容易添加到系统中，比如说要新添加一项功能，允许用户查询哪些食堂提供面条，则对于语言知识只需填加“面条”词条并设置其语义类为受事 THEME，但对于任务知识就需要考察校园内所有食堂的情况，再对数据库中所有食堂节点加上“卖-面条”的项目链。

错误处理过程所处理是用户知识与任务知识的不一致，处理方式是通过对语言知识来补充任务知识的不足。由于语言知识容易扩充，我们可以尽量加入超出当前任务知识表示的词条，以使其接近用户知识。这样，如果用户的询问触及系统功能局限或者超出场景，尽管不能在数据库中查询得到结果，我们仍然

可以利用语言知识所得到的理解给予一个恰当的应答。我们称用户询问内容在任务知识之外的，为超出任务问句（OOT, out of task）；称用户询问内容在语言知识之外的，为超出场景问句（OOS, out of scenario）；理想情况下，如果语言知识覆盖了整个领域，则在整个领域内有意义的问句都可以得到恰当的应答，超出其外的称为超出领域问句（OOD, out of domain）。如果语言知识范围覆盖了任务知识，则所有 OOS 都是 OOT。还有两种属于 OOS 的语言知识不完善的情况，超出词表（OOV, out of vocabulary）和超出语法（OOG, out of grammar）。

一般性的语言知识是人-人对话过程中知识共享的桥梁。只有一般性的语言知识，才能描述超出自身任务知识之外的语义信息。

## 第三章 句法分析

### 3.1 预处理

先介绍一下句法分析之前的两个步骤：分词和词性标注。

由于汉语字与字之间没有明显的词边界，而词又是表达意义的基本单位，因此分词成为汉语语言理解的第一个步骤。规则方法和统计方法都可用于分词。

EasyNav 第一版[42]中使用统计方法，利用 N-Gram 概率进行搜索，搜索策略类似于语音识别时的音节同步算法，只不过改成了词同步。这样，分词使用了和语音识别相同的词表和语言模型，即大小为 50624 的词表，和从 2 亿字语料中统计得到的 Tri-Gram 模型。分词首选准确率近乎 100%，因此可以直接使用分词首选进行后续处理。

第二版考虑到场景中几乎没有分词歧义，为降低时空开销，作了简化。词表改为数据库中和语料中出现的词集合，大小为 709。分词策略也简化为最长匹配。

词性标注步骤将词序列转化成为词性序列。规则方法和统计方法都可用于词性标注，我们选用统计方法[42]。利用东北大学提供的词同现频度统计，我们统计得到词性间的 N-Gram，和词类属概率（即词对应词性的概率）。根据贝叶斯公式，可以由词性 N-Gram 概率和词类属概率计算出词性序列的概率。由于统计基于的是一般性语料，直接使用进行词性标注准确度达不到很高，根据场景中实际的词性出现频度手工作了调整。以词性序列概率高于某个相对阈值为标准，词性标注输出多个词性标注序列。

### 3.2 句法规则

句法规则使用上下文无关文法 (Context-Free Grammar, CFG)，共计 191 条。从 91 个词性中经挑选、合并得到 32 个词类，对应于终结符。非终结符有 57 个，包括标准短语如 NP、VP、PP，次一级的短语如 NP+、VP++ 以及句法标记如 SUB、PRED、OBJ。更详细的介绍见附录 A。

句法规则来自于汉语语法[61]，但针对分析效率和场景特点作了调整。

第一，场景有限，删去不常见却会带来问题的规则。通常书面语中句子语法结构可以很复杂，如例(1)中主语是不相连复指成分，而复指成分又包括状语修饰的并列动宾短语，第二个动宾短语中宾语的名词定语又带有一个带状语并列动宾短语的定语，又如例(2)中带文言色彩的文体，出现许多不常见的单字词用法。这些在日常对话中很少出现，因此系统对相应规则不予支持。对于相对简单的句型和结构，在不会造成歧义的前提下，尽量加入相应规则，以使句法规则具有较大的覆盖度。如果会造成歧义但在场景中不常见，则删去。如在汉语语法中，主语可以是 NP、SP、VP 甚至单句 DJ，但在场景中，设定主语为 NP 和 DEJG 就够了；谓语可以是 NP、VP、AP、QP、PP，在场景中，设定 NP 不能作谓语。汉语语法具有的灵活性，使得在权衡语法覆盖度和准确度时，要比英语有更多的考虑。

例(1) 以某种方式参加推翻资本主义社会及其所建立的国家制度的事业，参加赖有他才第一次意识到本身的地位和要求，意识到本身解放条件的现代无产阶级的解放事业，——这实际上就是他毕生的使命。（恩格斯《在马克思幕前的讲话》）

例(2) 时下官场上确有不少官僚主义、形式主义的问题亟待整治，浮华、庸俗、惟上、贪逸等等不良作风在一些地方几成通病，笔者愿就自己之观察一一述之。

第二，在规则中广泛使用次级短语，消除冗余生成。比如从  $n$  到 NP，中间就有 NM 和 NP+ 两级，以其中部分规则为例：（使用竖分割线“|”将左端相同的规则合并写在一行）

```
nm    ->n
np+   ->nm | attr np+
np     ->np+ | np+ c np | np+ np
```

这些规则规定了从名词到名词词组的生成次序，先是考虑名词与定语的结合，再考虑并列的名词词组。并列名词词组的生成规则中也规定了从左端递归生成。如果没有定义次级短语分级书写生成规则，则可能会出现冗余生成。如：attr n n 不仅分析得到 [[attr n] n]，也得到 [attr [n n]]。又如 n n n 不仅分析得到 [[n n] n]，也得到 [n [n n]]。这种冗余生成会大大增加分析器的负担，降低计算效率。冗余生成带来的歧义是不必要的歧义，对于后续理解没有影响。

第三，使用低分辨率的规则，不去分辨对后续处理没影响的细节。如 *n* 和 *n* 结合成 NP，可以是并列结构，也可以是偏正结构，但具体是什么结构，对系统中后续的语义分析没有贡献，因此不对短语内部的结构进行分辨，只需要保证短语能对应正确的句法标记。又如例(3)是两个单句，且第二个单句省略了主语，系统为处理方便，把它看作一整句，但由于“去大礼堂”和“怎么走”都标注了正确的句法标记，对后续语义分析没有影响。

例(3) 我想去大礼堂怎么走？

第四，加入支持特殊句型的规则。如主题句例(4)中“自行车”是动词“修”的受事，宾语前置倒装。

例(4) 自行车哪有修的？

不需要复杂的机制，只要加入新的规则就可以进行支持。增加规则（只列出部分规则）：

*robj* → *np*

*djvh* → *sub vhp*

*dj\** → *robj djvh*

其中 *DJ\** 是次一级的单句，*ROBJ* 是倒装的宾语，*DJVH* 是以动词“有”为主动词的单句，*VHP* 是动词“有”和的字结构构成的短语。

第五，将一些经过词法变换得到的短语收入词表。如“知不知道”。一些具有一定规则性的地名，如“十一号楼”、“二十一号楼”也都收入词表。如果专门为它们写规则形式上会简洁些，但考虑到因此而需要把词类标记做到短语一层，还是直接收入词表方便一些。

由于当前的技术水平无法同时达到较好的覆盖率和准确率，句法规则的书写需要维持覆盖率和准确率的平衡。用词性作终结符，有很好的覆盖率，但准确率不高。用语义类作终结符，准确率很高，但规则想达到较好的覆盖率靠手工书写比较困难。用属性词典[8]考察词与词之间的组合关系，能在保证语法一般性的前提下提高准确率，但需要做很多对场景而言不必要的工作。最终系统中还是使用基于词性组合关系的 CFG，这是因为一方面，在尽量覆盖场景中句型的前提下，利用口语和场景特点降低覆盖率以提高准确率；另一方面，对于无法解决的歧义，留给语义分析去分辨。

### 3.3 GLR 分析器

句法分析使用 GLR 分析器，是 Tomita 在 LR 分析器的基础上提出的[62]。LR 分析器[63]是确定性分析器，特点是一遍搜索，能够很早就发现错误，缺点是只能处理 CFG 的一个子集：LR 文法，没有处理带歧义文法的能力，因而也就没有处理自然语言文法的能力。GLR 分析器针对这一点，提出了解决冲突的机制，以支持一般的自然语言文法。在 LR 表中表示冲突项是第一步，还需要能够解决冲突而带来的歧义。

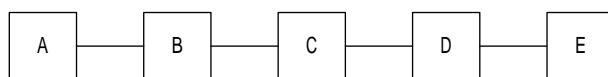
GLR 分析器使用图结构栈代替原先的线性栈，以模拟不确定性。图结构栈有三个主要操作：分裂、组合、局部歧义压缩。以下述生成规则为例：

规则 F： $F \rightarrow DE$

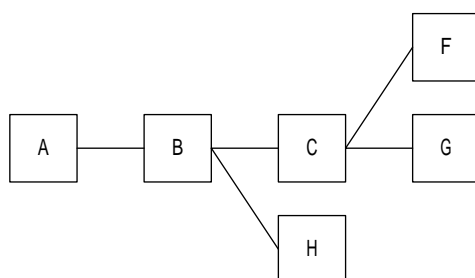
规则 G： $G \rightarrow DE$

规则 H： $H \rightarrow CDE$

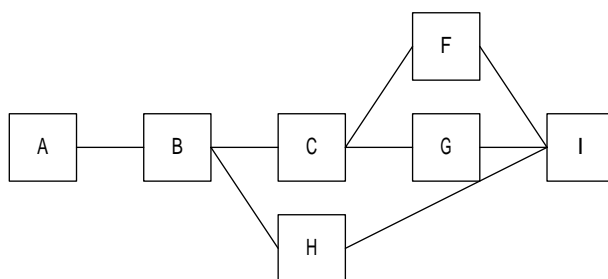
假设当前栈内容如下所示，从左到右，最左边的 A 在栈底，最右边的 E 在栈顶。



此时由于三条生成规则都符合，出现归约冲突，栈对应多种归约可能，分裂操作允许栈顶分裂以表示这多种可能。并行地使用规则 F、规则 G、规则 H，得到三个栈顶：F、G 和 H。



此时将 I 移入，尽管原先有多个栈顶，移入后经过组合操作，得到单个栈顶：I。

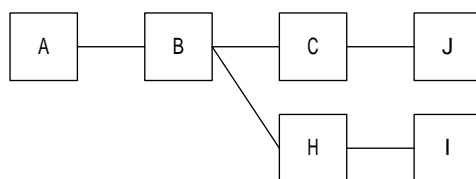


假设我们有如下规则：

规则 J1 :  $J \rightarrow F I$

规则 J2 :  $J \rightarrow G I$

此时规则 J1 和规则 J2 都符合，进行归约的结果得到两条相同的分支 A-B-C-J。局部歧义压缩操作将相同的分支进行合并，看作单一分支。在进行上述规则 J1 和规则 J2 的归约，以及局部歧义压缩操作后，栈变为：



对于带歧义的文法，输入句子可能会对应多棵分析树，压缩共享分析森林提供了一种将多棵分析树统一表示的方法。如果分析树之间拥有公共的子树，分析森林中只表示一次，这称为子树共享。如果子树的顶节点对应于局部歧义压缩的栈节点，则将多个子树的顶节点合并为一个压缩节点。

以下述具有歧义的句子为例：

例(5a) I saw [Jane with a telescope]

例(5b) [I saw [Jane] with a telescope]

对应于图 3-1 所示的分析森林：

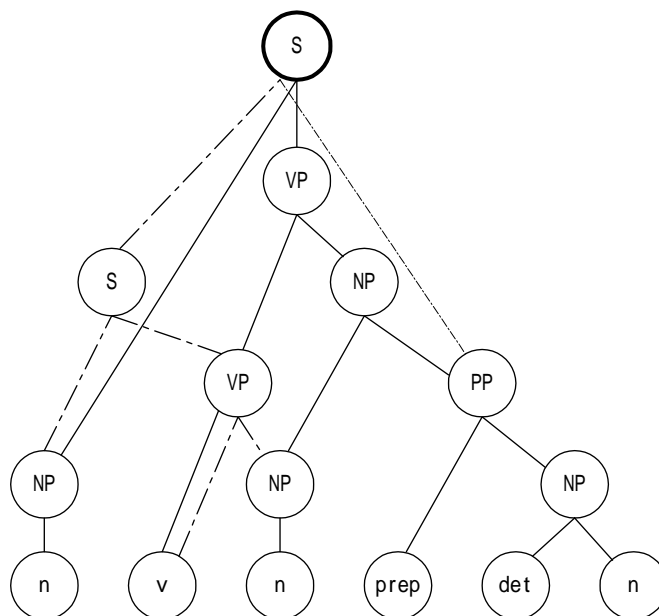


图 3-1：压缩共享分析森林示例

加黑的 S 节点是一个压缩节点，对应于两棵子树，分别用虚线和实线表示。两棵子树间有若干共享子树，如 PP 节点对应的子树，只表示一次，由两棵子树共享。

分析器所使用的分析表通常有 LR0、LR1、SLR1 和 LALR1 几种[63]。LALR1 分析表的描述能力强、表示效率高，因此系统中 GLR 分析器使用 LALR1 分析表。

我们在 EasyNav 系统中使用 GLR 分析器作为句法分析器，主要因为它具有以下优点：

第一，预编译，效率高，运行时消耗资源少。

第二，从左向右，与语音识别流程相似，可以并行进行以提高效率。

第三，图结构栈与语音识别输出的词网格结构相似，有可能直接以词网格为输入。

第四，对移入-归约操作中的歧义可以用概率描述，以便将来引入统计方法。

3.4 句法树和句法约束条件

句法分析的输出是压缩共享森林，展开后即为若干句法树的候选。在显示句法树时，为使树型结构更加清晰，将次级短语进行了适当的缩略。比如从 *n* 到 *NP*，要经过 *n* -> *NM* -> *NP+* -> *NP* 三步，中间的 *NM* 和 *NP+* 在句法树中都略去。比如 *n n n* 构成的 *NP* 短语，通过递归规则得到的是一个多层的结构，缩略后则成为一个单层的结构。因为次级短语本来就是为了解析效率而引入的符号，所以在显示时缩略，不管对于人直观上的理解，还是对于系统后续的语义理解，都不会有影响。

语义分析所使用的并不是整个句法树，而是从其中抽取出的句法约束条件。图 3-2 显示了一个句法树的例子，用黑体标出的是句法标记。通过向上搜寻词的祖先节点，我们可以得到词所对应的句法标记串，也即语义分析中使用的句法约束条件。通过句法标记串，我们不仅可以确定词在句子结构中所处的地位，而且可以确定词与词之间的句法关系。如图 3-2 中，“哪”对应的句法标记串为 SUB-DJ，“吃饭”对应 ATTR-OBJ-PRED-DJ，若已知“哪”填在槽 1 中，也可以写成 ATTR-OBJ-PRED-[F1]，又若已知“地方”填在槽 1 中，则可写成 ATTR-[L1]。其中[L1]对应于词的句法标记节点，[F1]对应于[L1]的上一级句法成分祖先节点，[C1]指槽 1 对应的句法标记，统称为引用标记。使用这些标记，可以描述出动词的施事受事、修饰成分、并列结构等句法约束条件。

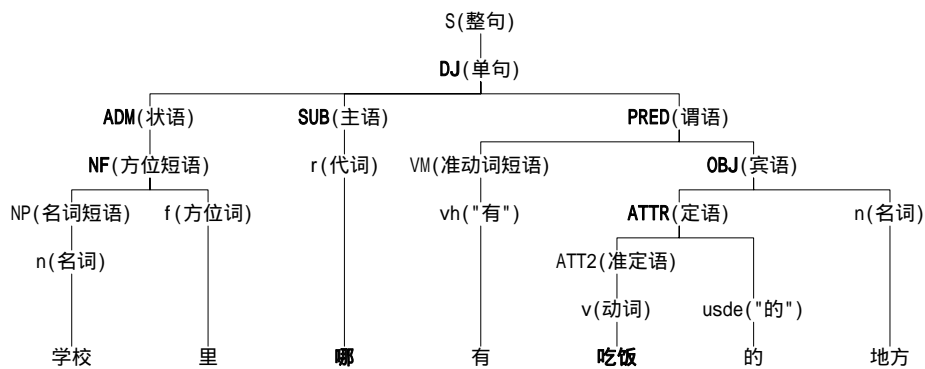


图 3-2：句法树示例（句法标记和关键词用黑体标出）

从这里也可以看到前述不必要的歧义对句法约束条件没有影响。句法约束条件考察的是句法标记的层次关系，次级短语的结合次序乃至结构关系对此没

有影响，也就不会对后续语义分析造成影响，这就是前面规则书写原则考虑的基准。

句法约束条件的引入使得理解的中心由句法分析转向语义分析，句法约束条件在语义规则中定义，句法规则的修改与完善对于句法约束条件的抽取基本没有影响。这样，句法分析模块可以独立地完善，增补规则、加属性约束条件、改变词类集合基本上都不需要改变语义分析模块和语义规则，从而使得系统易于维护与扩展。采用一般性的句法知识，也使系统容易移植。

### 3.5 句法分析的作用

句法分析通常起两个作用，一是筛选，即以符合语法作为句子合法性的判断，二是理解，即提供句子结构信息为进一步得到语义信息服务。在口语对话系统中，认为用户友好，不会故意说不合语法的句子，筛选不是必要；领域受限，理解对于句子结构信息的依赖不强。这种情况下，许多系统并不使用句法分析，而是直接从检出的关键词或词组序列映射到用户意图，这么做还可以增强对语音识别输出的鲁棒性。

尽管如此，我们认为句法分析在口语对话系统的语言理解过程中还是有意义的。

首先，必须承认句子具有结构，句子意义并不是词义的拼凑。对简单句子，不使用句法知识，仅用词义的拼凑也难以解决问题。如从到路线句中有两个地点词，则哪个是源地点，哪个是目的地点，如果没有句法分析，就需要给予特别的处理规则去利用结构信息。不使用句法知识，必然就会走上使用经验主义的非系统办法解决的方向，而对自然语言这样复杂的系统而言，无法想象不使用系统的办法最终能解决。

其次，句法知识除了为语义分析提供约束之外，对上下文分析和应答生成也都有一定作用。指代分析倾向选择要考察引用项的级别，很大程度上依赖于引用项的句法位置。省略分析所处理的省略现象，也有相当一部分依赖于对特定句法结构的辨认。应答生成参考输入句子的句法结构，做出与询问句型接近的回答，会令用户感觉更自然。

另外，句法知识可能能对语音识别搜索起到剪枝效果。使用基于统计的N-Gram作为语言模型是当前语音识别获得成功的重要原因，不过考虑到口语对

话系统场景有限，获取大规模语料不是很容易，句法知识对于帮助建立语音识别所需的语言模型可以起一定的作用。

不过我们还是弱化了句法分析的作用，以削弱理解对于句子结构信息的依赖程度。句法分析和语义分析的处理过程相对独立，各自完成各自的功能。句法分析实现语法合法性的筛选功能，并抽取句子的结构信息。语义分析在综合包括上下文在内的各种知识的基础上实现理解功能。句法分析抽取出的结构信息也被语义分析使用，但并不作为语义分析的基础。事实上，语义分析对词标注语义类后又重新进行了一次生成过程，而不是将已有的句法结构转换为语义结构。语义分析所使用的只是句法约束条件，而需要哪些句法约束条件，句法约束条件要约束到什么程度，也都是由语义规则定义的，不受句法规则的影响。

这种句法分析和语义分析相对独立的分工，使得语言理解模块有相当的灵活性。汉语语法比英语灵活，写一套覆盖率和准确率都很好的规则系统不很容易。语音识别结果中经常有词识别错误，也会使得句法分析效果下降。理解由语义分析模块完成，避免受句法分析不可靠结果的影响。句法分析和语义分析两个模块功能上的独立，使得两个模块可以独立地完善，方便了系统的开发。

## 第四章 语义分析与表示

### 4.1 语义分析的四个层面和意义组合原则

正如在引言中所指出的，语义分析与表示是自然语言理解中的重要问题，也是一个难题。口语对话系统中的语义分析与一般性自然语言理解相比，更偏重研究用户想要做什么，即进行言语行为及其相关信息的分析。

在口语对话系统背景下考察语义分析，根据其要解决问题的难度，有四个不同层面的要求：

第一是模式划分，确定句子对应的具体言语行为属于哪一类。如 Lucent 的 NLCR 系统[39]，电话路由只需要确定路由目的地，而不需要分析进一步的语义信息，是典型的模式划分问题。

第二是结构表示，确定句子对应言语行为的语义结构。如 LIMSI 的 RailTel 系统，在进行模式划分确定言语行为后，再将相关信息填入该言语行为对应的语义框架中[3]。又如 MIT-SLS 的 Mercury 系统，先通过句法分析得到句子的句法结构，再进行转换得到语义结构[64]。自然语言能够生成的句子是无限的，但句子可能的结构是有限的，对于较复杂的语义理解问题，模式划分难以解决，需要结构表示才能解决。

第三是意义计算，确定句子对应言语行为的含义。理解需要的并不是句子的语义结构，而是句子含义的物理解释。以对话系统中常见的言语行为如查询请求、命令请求为例，意义计算将查询请求转换为数据库查询语言，将命令请求转换为操作命令序列。如 Mercury 系统，使用转换规则分析语义表示，生成 SQL 语句[64]。

第四是上下文处理，确定句子在上下文中的含义。在口语对话过程中，不仅要能理解单句请求的意义，还要能理解连续对话中隐含了若干信息的请求的意义。语义表示中必须保留有并入上下文信息的空间，语义分析也需要考虑包括上下文处理的统一流程。如 MIT-SLS 的 Voyager 系统[23]，要处理当前位置和焦点位置的指代。又如 RailTel 系统，在进行查询请求时要利用之前若干句中所保留的信息[15]。

以上四个层面呈现一种递进关系，后一层面的视野要高于前一层面，对应于更复杂的语言理解要求。同时基于后一层面的应用通常也包含了之前的各个层面。语言理解任务难度越高，其语义分析所需要达到的层面也就越高。

意义组合原则[60]是语义分析应当遵守的一个准则。自然语言具有无限的生成能力，但可能生成的结构类型是有限的。进一步的，自然语言所能表达的语义也是无限的，但语义结构类型是有限的。意义组合原则指出，为描述无限生成的语义能力，整句的意义必须通过由底向上地依据句法结构组合句中各个语义成分的意义来得到。

意义组合原则对语义分析的后三个层面都有影响。结构表示层为符合该原则，不应使用预定义的方式来获取语义结构，而要通过将低级语义成分组合成高级成分的过程获取语义结构。意义计算层也不应在语义结构得到后再进行翻译得出其含义的物理理解，而要在语义结构生成的组合过程中同步进行意义计算。上下文处理层也需要在本句语义成分的意义组合过程中适时考虑上下文语义信息的加入。

对语义分析进行模式划分、结构表示、意义计算和上下文处理四个层面的划分，虽然是本文在口语对话系统背景下提出的，但从一般性自然语言理解的大方向上来看，这一划分也是合理的。

下文中所讨论的各种语义分析方法，以这四个层面的划分以及意义组合原则作为基础评价各自特点。另外，EasyNav 场景单回合交互的特点要求对用户输入的复杂句子有分析和表示的能力，这也是一个评价指标。

## 4.2 句型模板方法

EasyNav 前后两个版本中语义分析采用了不同的方法，第一版使用的是句型模板方法，第二版使用的是互连框架方法。这两个方法都是以其所基于的语义表示而命名的。本节先介绍第一版所使用的句型模板方法。

### 4.2.1 整句框架与句型模板

第一版[42]中使用单个语义框架作为整句的语义表示，我们称之为整句框架。我们定义了一系列句型模板，每个模板中包含了若干槽，类别槽指示了问

句类型，如 WHERE、WHICH、HOW-LONG 等等，其它槽填入字串，字串来自句中符合槽约束条件的关键词。与输入句子进行模板匹配后，得到的句型模板直接生成语义框架。图 4-1 是一个整句框架的例子。

句型模板号：6
类别：where
施事：学生
限制1：计算机系
功能：住
距离：最近
受事：空
限制2：空

图 4-1：整句框架示例

整句框架的特点是以句子为表示单位，一个框架包含了句子中的所有细节。通常的语义框架中槽内容可以是值，也可以是框架，从而形成一个嵌套结构，有很强的表示能力。整句框架槽的内容只能为值，是对语义框架的一个简化。

由于整句框架的表示基于句型而非句子中的语义对象，后续模块使用这一语义表示的时候，也只能根据不同句型采取不同操作，而没有一个统一的处理流程。框架内的槽之间没有显式的关联，只有在具体的处理流程中才表现出这些语义要素间的联系。

在针对不同句型采取的不同处理流程中，整句框架转换成查询框架。查询框架是一个更为统一的语义表示形式，不同的句型都要映射到同一个查询框架。但查询框架仍以整句为单位，框架内槽之间依然没有关联，因此查询模块在使用查询框架时仍需对如何查询进行解释。不论整句框架还是查询框架，都没有嵌套表示的能力。对应的，查询也只是对有限的查询句型进行预定义的解释流程。

#### 4.2.2 句型模板规则

第一版的语义分析使用带有句法约束条件的句型模板规则。

句型模板定义了 5 类, 共 51 个, 这五类分别是询问 1) 在哪里 (WHERE); 2) 是哪个 (WHICH); 3) 怎么走 (HOW-TO); 4) 有多远 (HOW-FAR); 5) 有多久 (HOW-LONG)。图 4-2 给出了部分句型模板的列表及图示。

句型模板1			
类别: WHERE	槽1: 地点名 语义类: LOC 句法约束条件: SUB OBJ		1: where + LOC( sub, obj )
句型模板5			2: where + gLOC( sub, obj )
类别: WHERE	槽1: 机构名 语义类: OFFC 句法约束条件: SUB OBJ	槽2: 地点类名 <可选> 语义类: gLOC 句法约束条件: [L1]	3: where + gLOC( sub, obj ) + DIST( attr-[l1], pred, dejg, adm, nf-adm )
句型模板42			4: where + gLOC( sub, obj ) + SPEC( attr-[l1], [l1] )
类别: WHICH	槽1: 地点类名 语义类: gLOC 句法约束条件: SUB OBJ	槽2: 地点属性 语义类: SPEC 句法约束条件: ATTR-[L1] PRED DEJG	5: where + OFFC( sub, obj ) + [gLOC]( [l1] )
			6: where + [SUB]( sub ) + [SPEC]( attr-[l1], [l1] ) + FUNC( pred, attr-sub, attr-obj, dejg ) + [DIST]( [c3]-[f3], pred, dejg, adm, nf-adm ) + [OBJ]( obj-[l3] ) + [SPEC]( attr-[l5], [l5] )
			...
			42: which + gLOC( sub, obj ) + SPEC( attr-[l1], pred, dejg )
			...

图 4-2: 句型模板列表及图示

模板由一个类别标示和若干个槽构成, 规则中槽之间用“+”号隔开。语义类上加上中括号的, 在图中对应有一个“可选”标志, 表明该槽未被填入关键词模板仍可被匹配。句法约束条件之间用“,”隔开, 是“或”的关系, 只要有一项条件满足就可将词填入槽中。各个槽中填入的语义要素之间存在的语义关系(如: 动作与其施事受事、物品与其修饰成分等), 通过带引用标记的句法约

束条件表现出来。

### 4.2.3 模板匹配过程

由于存在语义要素之间的句法约束条件，模板匹配是一个多遍分析过程。带引用标记的槽在所引用槽未填入词前无法判断句法约束条件是否满足，因而首先要进行一遍不带引用标记的槽的匹配，再对带引用标记的槽进行匹配。后一匹配可能要进行多遍，直到不再匹配为止。比如需要先匹配上功能动词 (FUNC)，才能去匹配受事 (THEME)，之后才能去匹配受事的限定词 (SPEC)。流程如图 4-3 所示。语义类到模板的链接表中，句法约束条件中带引用标记的对应有效链接表，不带引用标记的对应休眠链接表。当休眠链接的引用标记所指向的槽被填充后，该链接成为活动状态，可以向该链接对应的模板槽填充关键词。

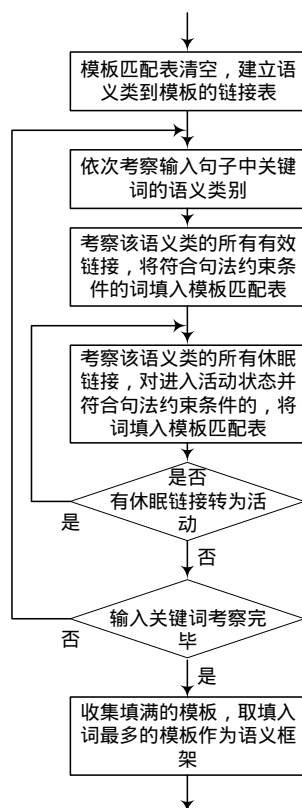


图 4-3：模板匹配流程图

句型模板匹配以意图理解为目标。句型模板对应于围绕查询动作（疑问词）的语义框架，框架中的槽如地点名、参照地点名、地点属性等，都是为完成查询操作所需要的语义信息，一个模板号对应于一个意图。句型模板匹配方法的优点在于：

- 1) 基本不依赖于句法分析；
- 2) 简化了语义分析处理的流程；
- 3) 模板匹配准确、唯一，能消除句法分析带来的歧义；
- 4) 直接分析用户意图，得出语用信息，如例(6)所示；

例(6) 你知不知道哪有食堂？

显然，对例(6)不应该回答“是否知道”，而应回答某一食堂的位置。作为友好的应答，还应回答这一食堂的名称。这些语用信息蕴含在场景中，难以用通用的办法描述。这也是语义分析将整句模板作为一个整体去把握，而不是从局部望文生义去理解的原因。

从以上优点可以看到，使用句型模板匹配方法的语义分析是 *EasyNav* 系统中理解的中心，句法信息只是语义分析所考察信息中的一个方面。以意图理解为目标，抽取与查询意图相关的语义信息，辅以场景知识，针对性强，消耗资源少，符合有限场景对话的要求。场景相关的句型模板定义规则与分析模块分离，规则只需考虑查询相关信息而较易书写，这都使得系统易于维护与扩展。

### 4.2.4 方法评价

从语义分析四个层面划分的角度来看，句型模板方法的层面是比较低的。根据整句句型进行分类，属于模式划分层面。生成整句框架和查询框架，属于结构表示层面，这一结构表示不满足意义组合原则。不包含意义计算层面，意义计算由后续查询模块来做，由于查询模块得不到什么计算信息，只能通过预定义方式进行映射，也不可能满足意义组合原则。不包含上下文处理层面。

从分析和表示复杂句子的角度来看，句型模板可以表示复杂句子，但表示效率比较低。缺乏机制消除相似句型模板中的冗余信息。以整句为单位建立模板，也无法有效描述递归结构的句子。

## 4.3 互连框架方法

前面的两节只是准备部分，本节开始才进入了主体部分，介绍当前第二版中使用的语义分析[52]和表示方法[53]。

### 4.3.1 互连框架和主题结构

互连框架是主要的语义表示，查询链对应其内包含的计算信息。主题结构是为上下文分析服务的辅助性语义表示。

#### 4.3.1.1 互连框架

语义表示常见有两种方式：逻辑式和语义框架。逻辑式对应于形式化的命题推导，语义框架对应于程序内建的推理[50]。我们采取基于语义框架的方法，是因为对于特定领域特定任务的情形而言，推理由程序化系统完成更为有效与方便。

我们提出的互连框架和标准的语义框架有所不同。以 MIT 系列系统中的语义框架表示[65]为例，框架中的槽可以为值或者是框架，父框架中的槽包含了子框架，就形成一个嵌套的表示形式。互连框架则是由一组框架组成，框架间通过槽与槽之间的连接联系在一起。

互连框架中的槽有三种类型：值、指针、函数。值型槽赋予槽一个值。指针槽指向本框架或其它框架的一个槽，槽的值为该指针对应的查询链返回值。指向本框架槽的指针称为近指针或本地指针，指向其它框架槽的指针称为远指针。函数槽对应一个应答函数或查询函数，其值为函数返回值，函数参数为一组近指针。槽还可以附带前提条件，前提条件由一组远指针构成，用以约束指针槽或函数槽的返回值。

以描述范围属性的规则（规则 12）为例，该规则所生成的框架如图 4-4 所示：

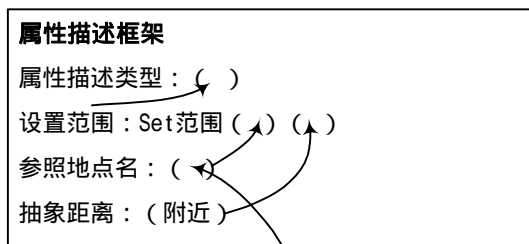


图 4-4：框架示例

互连框架具有以下几个优点：

第一，语义框架只是描述了语义层次结构，互连框架不但包含了语义信息，还包含了如何获取这些语义信息的计算方法。举例来说，“九号楼”对应的地点框架包含地点名和地点位置两项信息，但有多种说法都对应这样的地点框架：

例(7)

九号楼

计算机系学生宿舍

计算机系本科生住的地方

离八号楼最近的楼

十号楼南边那栋楼

.....

上述的短语都对应于地点框架，框架中的“地点名”槽包含有地点名信息，但除了第一个短语以外，这一信息都不能直接从文本中得到。互连框架将不能直接获取信息的槽设置成为指针槽或者函数槽，该槽指向的查询链定义了查询这一信息的方法。

第二，互连而非嵌套的形式，给予了互连框架很好的组合性，这种组合性对于上下文意义重构有重要的意义。在互连框架中，每个框架不仅包含了语义对象相关的信息，还包含了如何去获取这些信息的计算方法。这样，每个框架就成为自给自足的个体，尽管框架是在分析一个具体句子的过程中生成的，但仍然可以脱离整句话的语义表示而单独存在。基于互连框架的上下文分析过程，是从上下文中找到了一个语义结构，并与当前的语义结构相组合，而不是仅仅

从上下文中找到了一个词。

第三，框架内容是动态生成的，框架之间也是动态互连的。以例(7)中“九号楼”不同说法对应的地点框架为例，地点名信息对应于不同的计算方法，因此不能预先定义框架格式直接生成，而是要在语义分析过程中动态生成，并将“地点名”槽与动态生成的其它框架动态互连。这种动态性是面向自然语言的语义表示所必需的灵活性，具备了这种灵活性，互连框架就有了表示嵌套或多层语义结构的能力，而不只是有限的几种句型。

综合上述优点可以看到，互连框架是以语义框架为基础发展出来的表示方法，面向一般性的语义表示，由于本身包含了信息的计算方法，而具有很好的组合性，特别地，这种组合性符合上下文分析的需要。

#### 4.3.1.2 查询链

“意义是什么”是语义研究的重要问题。从意义的不同侧面考察，有指称义、真值义等不同观点[66]。我们认为，问句的意义是查询义，即描述了什么要查询的查询表达式。查询链就是描述查询义的语义表示工具。

在互连框架中，当槽的类型为指针或函数时，槽的值不能直接得到。对于指针槽，值为指针所指向的槽的值；对于函数槽，要先从参数指针指向的槽获取函数参数，再计算函数值；如果有前提条件，还需要先根据前提条件设置限制，再进行计算。槽值的获得是一个递归过程，计算一个槽的值需要根据其指针的链接先计算其它槽的值，直至链接到直接有值的值型槽。将槽间链接的总体定义为查询链，我们认为槽的值可以通过槽所指向的查询链获得。

特别地，对于意图框架的应答内容槽所指向的查询链，我们称之为应答查询链，该槽称为应答查询链的入口。应答查询链对应于整个问句查询义的计算，根据应答查询链可以查询得到用户所需信息，并给予合适应答。应答查询链的入口如果只是一个指针槽，则直接以查询链的返回结果作为应答；如果是一个函数槽，则该函数作为一个应答模板函数，由程序根据其参数返回的值组织应答。

图 4-5 给出了一个地点名应答槽指向的应答查询链的例子。

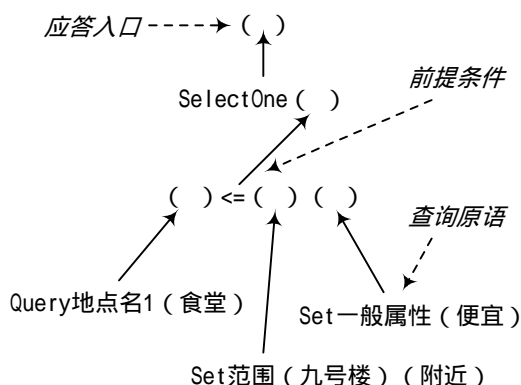


图 4-5：应答查询链图示

查询链先链接到地点框架的地点名槽，其值的计算定义为对地点名集合槽的值进行多选一操作 `SelectOne`。地点名集合槽值的计算又定义为带两个前提条件限制的指针。指针指向地点类框架中的地点名集合槽，其值定义为根据地点类“食堂”查询得到的地点名集合。第一个前提条件指向范围框架的设置范围槽，限制前述查询需在范围“九号楼附近”内进行。第二个前提条件指向一般属性框架的一般属性槽，限制前述查询需在有性质“便宜”的地点集合内进行。

可以看到，应答查询链的基本单位是查询原语，查询原语以一定方式组合，构成整条应答查询链。查询原语的组合构成查询操作，如图中的 `Query 地点名 1 (食堂)` `Set 范围 (九号楼) (附近)` `Set 一般属性 (便宜)`，三个查询原语组合构成一个对地点名集合的查询操作，前提条件的引入提供了将复杂查询函数拆解成查询原语的方法。

查询原语不是最基本的查询单位，但却是基本的查询义表示单位。以查询原语而非查询函数为单位描述查询义，一个原因是简洁、避免冗余，但更重要的原因是查询原语可以直接与语义对象相对应，而查询函数却不能。当语义分析将范围对象、性质对象和地点类对象组合生成一个地点对象时，对象内部的范围查询原语、性质查询原语和地点类查询原语也组合起来，构成一个描述完整查询操作的查询链。

前提条件的引入提供了查询函数拆解的方法，从而使查询义也能够依据意义组合原则生成。每个语义对象的查询义对应于框架中槽所指向的查询链，对象的组合对应框架互连，并导致查询链的链接。查询义的组合性，也使得查询模块具备了动态查询组织的能力。

框架中的槽包含了语义信息，槽指向的查询链则定义了语义信息的计算方法。互连框架内含的应答查询链提供了用户所需语义信息的计算方法，适合作为意图的表示方式。

#### 4.3.1.3 主题结构

主题结构所表示的是关注状态，其作用是描述会话过程中可能被下文引用的语义对象。尽管是语义表示，但其目的不是语义理解，而是上下文维护。此外，针对单回合交互模式的 EasyNav 场景中主题切换频繁的现象，主题结构所起的另一作用是描述当前主题中的对象、属性和关系，以便在此基础上研究主题变化的条件和对引用的影响。

在图 4-6 所示的主题结构图中，节点表示主题项目，有向弧表示主题项目间的关系，被有向弧连接的起止节点相应称为父项目和子项目。

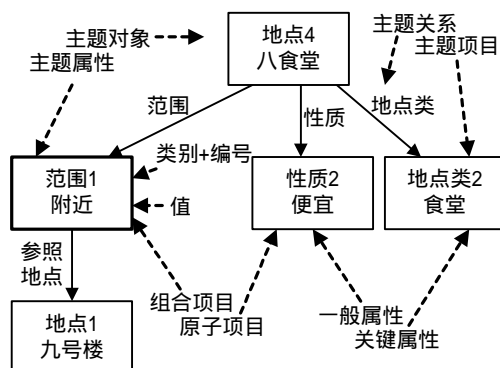


图 4-6：主题结构图示

在 EasyNav 场景中，共有 12 种主题项目：地点、地点类、机构、方位、方式、性质、最值条件、施事、受事、功能、范围、路线。这些项目又分为原子项目和组合项目两类。前七种属于原子项目，其本身就代表了完整的含义。后五种属于组合项目，需要结合其子节点内容才能得到完整的含义，才能对应于一个确定的引用项，其中施事、受事、功能项目也可以退化成为原子项目。在引用项拷贝时，原子项目只需拷贝单个主题项目，组合项目则需要连同其子项目一同拷贝。

在主题结构中，没有父项目的顶层项目为对象，如地点 4；父项目为组合项目的子项目可作为对象，也可作为属性，如地点 1；父项目是原子项目的子项

目为属性，如范围 1、性质 2 和地点类 2。地点类和功能为关键属性，其余为一般属性，属性间具有优先级。在指代匹配时，作为对象被引用和作为属性被引用有不同的匹配条件，属性引用需要考察属性与父项目间的主题关系是否匹配。在省略校验时，需要依据父项目的所有属性综合判断主题变化情况，其中关键属性的存在与否将决定主题是否延续，优先级则在主题维持时决定哪些属性允许属性延续，哪些不允许。

主题结构是为上下文分析服务的语义表示，这决定了主题项目的定义原则。主题项目与客观世界上的实体相对应，这一对应关系由赋予项目的编号所定义，而非项目的名字（即：值）。主题类别不同于有层次继承关系的语义类，主题类别中只包含基类信息，而认为子类信息本身是额外的、可以被单独引用的实体。如图 4-6 中作为地点 4 的八食堂，其主题类别是地点，而不是食堂，子类信息作为属性，又单独构成了一个主题项目地点类 2。下文中可能引用地点 4，也有可能单独引用地点类 2。

此外，对语义理解有贡献、但对上下文没有影响的语义对象，也不能成为主题项目，比如说查询相关的语义对象。查询的是地点名还是地点位置，属于意图信息，查询该如何进行，属于意义计算信息，都不在主题结构表示的范围之内。因此，尽管主题结构形式上类似语义分析树和语义网络，但本质上并不相同。主题结构所关心的不是整体结构，而是单个元素，以及该元素所处上下文对其引用的影响。

主题结构不是预定义的，而是在语义分析过程中动态生成的。在单主题场景中，完成任务所需要的信息可以罗列，就好像是一张包含若干固定项目的表单，对话是在这张表单的上下文中进行的，主要需求是根据上下文控制对话进程。EasyNav 场景并不是这种上下文预定义且不发生主题切换的场景，单回合交互模式使得用户在一句中提供足够信息，这些信息的组合情况很多，无法预先定义。主题转换时要保留哪些信息，也难以一一枚举。依据意义组合原则动态生成主题结构，才能满足多主题场景的要求。

语义理解过程中，互连框架是主要的语义表示，但主题结构也起一定的辅助作用。主题项目和框架槽间有一一对应的关系。当语义分析将当前句的语义对象与上下文中的语义对象进行组合时，要把表示语义对象的框架互相关联起来，即对相应框架建立槽间指针，槽间指针的定位需要主题结构的帮助。应答入口是整句查询义计算入口，主题项目指向的框架槽则是相应语义对象的查询义计算入口。

由于主题结构的主要功能是为上下文分析服务，本章的介绍相对有限，下一章中还会对主题结构进行更多的讨论。

### 4.3.2 语义规则

在介绍各个语义表示之前，先介绍一下用于生成这些语义表示的语义规则。

举描述范围属性的规则 12 为例：

// 规则 12

属性描述->地点(.) + 附近(NEARBY)

槽：[属性描述类型]=设置范围

槽：[设置范围]=Set 范围( [参照地点名], [抽象距离] )

槽：[参照地点名]={地点}.[地点名]

槽：[抽象距离]=<NEARBY>

关联：[抽象距离]>> 范围 ^范围

关联：地点 >> 范围 ^参照地点

省略：地点 ~地点 >> 范围 ^参照地点 =

除去注释的第一行为**框架生成规则**，用于定义从已有框架（或者说对象）如何生成新框架（新对象）。规则采用 CFG 的形式，不过并没有句法规则中要求的词序和邻接限制。“->”分隔开规则左右两端，规则右端由“+”分隔开若干语义项，语义项分子框架（非终结符）和终结符两种。规则中语义项都用中文，括号中是备注。“地点(.)”中的“.”标明地点项是可省项，还有“\*”用于标明可选项。“附近(NEARBY)”中的“NEARBY”表明附近项是终结符，对应于语义类为 NEARBY 的语义要素（或者说关键词）。如果既是省略项，又是终结符，则同时列出，用“,”隔开。整条规则表明可以由地点对象和附近符号生成一个属性描述对象，也即，生成一个属性描述框架。

以“槽：”打头的是**槽赋值操作**，指示框架中槽的内容。用“=”分隔开槽名称和槽的内容，左端槽名称按习惯加上方括号。槽内容有值、指针、函数三种情况。“抽象距离”槽就是值型槽，槽的值直接取框架生成规则中附近项对应

的关键词。“属性描述类型”和“参照地点名”两个槽都是指针槽，前一个为近指针，后一个为远指针。近指针指向本框架中的槽，通常要加方括号，但表示类型的槽按习惯不加方括号。远指针指向其它框架中的槽，由用花括号括起的框架名和方括号括起的槽名构成，框架名对应于规则右端的语义项之一。“设置范围”槽是函数槽，函数的参数是一组近指针。

对于指针槽和函数槽，还可以附带一个前提条件。规则 8 是一个前提条件的例子：

// 规则 8

功能描述->动作(ACT,.)+ 施事者

槽：[地点名集合]=Query 地点名 3([动作])<={施事者}.[施事类型]

槽：[动作]=<ACT>

关联：[动作]>> 功能 ^地点类

关联：施事者 >> 功能 ^施事

省略：ACT =

前提条件由一组远指针组成，其作用是为槽对应或是槽指针指向的查询函数提供前提条件。例如规则 8 中“地点名集合”槽是附带前提条件的函数槽，函数“Query 地点名 3”是要由给定动作查询对应地点集合，加上“施事者”的前提条件后就成为由给定动作和动作的施事来查询对应地点集合。

规则 8 中框架生成规则右端动作项的备注“(ACT,.)”就是上述既为可省项，又为终结符的例子。

以“关联：”打头的是**关联操作**，指示主题结构的生成和内部连接的建立。“>>”分隔符表示左右两端的符号建立联系，“^”表示右端为主题关系。以规则 8 为例，两条关联操作对应的功能可以用如下子操作表示：

new 功能，set 功能 [动作]，施事者 >> 功能 ^ 施事

功能 >> ?? ^ 地点类

也即，关联操作要完成三项子操作：1) 创建新的主题项目；2) 向主题项

目赋值；3) 将该项目与其它项目建立连接。具体到规则 8，关联操作首先创建了功能项目，并将“动作”槽与功能项目相关联，也就是将槽的值赋予项目，然后将“施事者”子框架对应的项目以“施事”关系作为子项目与功能项目连接，最后设置功能项目将与下一步归约中会生成的新项目以“地点类”为关系建立连接。规则中的关联操作在书写上作了压缩：新出现的项目必然要创建，不需要再明确指出；槽关联与项目关联统一用“>>”符号联系；“^”在左端未出现子项目的情况表明是右端项目与未知项目建立连接时的关系；左端的子项目可以有多个，还可加上一个赋值的槽，中间以“,”隔开。压缩后得到上述实际规则中的关联操作形式。

项目赋值所关联的槽不一定是值型槽，也可能是指针槽或函数槽，这时对项目赋的值要通过槽所对应的查询链进行查询后得到，如规则 23 的地点项目：

// 规则 23

场所代称 -> 机构名(OFFC)

槽：[地点名]=Query 地点名 2( [机构名], [场所描述] )

槽：[地点位置]=Query 地点位置( [地点名] )

槽：[机构名]=<OFFC>

槽：[场所描述]=GetDefault( [机构名] )

关联：[机构名] >> 机构 ^机构

关联：机构, [地点名] >> 地点

地点项目所关联的“地点名”槽是个函数槽，则项目的值要在使用函数“Query 地点名 2”进行查询后才能得到。

规则 23 是单条规则生成多个主题项目的例子。框架和主题项目间有一对一的对应关系，每个框架对应一个主题项目且各自不同，每个主题项目对应一个框架中的槽，多个项目可能对应同一框架中的不同槽，正如规则 23 中机构项目和地点项目分别对应于同一框架中的“机构名”槽和“地点名”槽。在框架中包含若干项目的情况下，根据项目间的关系选择顶层项目作为框架对应项目，在本例中就是地点项目。在下一步归约的关联操作中，如果需要场所代称框架对应的项目，所对应的就是这一地点项目。

项目赋值的特殊情况是将项目与两个槽相关联。如规则 27：

// 规则 27

地点类描述 -> 地点类(gLOC)

槽：[地点名集合]=Query 地点名 1( [地点类] )

槽：[地点类]=<gLOC>

关联：[地点类; 地点名集合]>> 地点类 ^地点类

规则中地点类项目与两个槽相关联，一个是值型槽，一个是函数槽，用“；”分隔开，我们称之为**双槽关联**。项目的赋值是取值型槽的值。之所以有双槽关联现象，是因为要满足省略分析中从上下文重构语义对象的需要，这在 5.3.3.7 中会详细介绍。规则 22 中也有双槽关联现象，甚至从查询链的角度框架中本来没有设计相应函数槽，为了项目关联需要，专门增加一个“同描述地点名集合”槽。

以“省略：”打头的是**省略项重构操作**，指示在发现省略后如何重构。重列一下规则 12 的省略项重构操作：

省略：地点 ~地点 >> 范围 ^参照地点 =

第一个“地点”对应于框架生成规则右端省略的语义项，“~”后的“地点”指示省略项的主题类别，“>> 范围 ^参照地点”表明省略项应该作为子项目以关系“参照地点”与范围项目相连接，“=”表明这是一个值型省略，如果不标“=”则通常是一个函数型省略。值型和函数型省略的讨论见后面的 5.3.3.6 节。

重列一下规则 8 中的省略项重构操作：

省略：ACT =

之所以没有“^”“>>”“^”等信息，是因为所省略的语义项不对应于完整的主题项目，而只用于对项目赋值。尽管“动作”项省略了，关联规则依然可以生成相应的主题结构，因此就不需要省略项重构操作再做同样的工作了。

可以看到，省略项重构操作的作用是在出现省略的时候，代替关联规则生成部分的主题结构，并与关联规则生成的主题结构连接在一起。这样尽管语义项被省略了，在主题结构中依然对它进行了表示，使得我们可以得到该省略项

的类别以及与其它主题项目的关系，这些信息在省略分析从上下文寻找匹配的引用项时得到利用。

// 规则 44

询问路线时间->从字短语(.) +到字短语(.) +方式(MODE, .) +多长时间  
(HOW\_LONG, .)

槽：[应答内容]=路线耗时

槽：[路线耗时]=Query 路线耗时( [源地点名], [目的地点名], [方式] )

槽：[源地点名]={从字短语}.[源地点名]

槽：[目的地点名]={到字短语}.[目的地点名]

槽：[方式]=<MODE>

关联：[方式]>> 方式 ^方式

关联：[应答内容], 从字短语, 到字短语, 方式 >>路线

省略：从字短语 ~地点 >> 路线 ^源地点=

省略：到字短语 ~地点 >> 路线 ^目的地点

省略：MODE ~方式 >> 路线 ^方式=

规则 44 右端四个语义项都是可以省略的，注意“多长时间”项是疑问词，在 EasyNav 的语义规则中疑问词大都可以省略。对疑问词省略采用特殊的处理步骤，不需要对应的省略项重构操作。

语义规则共有 64 条，其中，带省略项重构操作的语义规则有 16 条，带疑问词省略的语义规则有 21 条。

### 4.3.3 语义分析过程

第二版的语义分析主要基于互连框架的语义表示，我们称之为互连框架方法。前面已经介绍了语义分析所生成的语义表示和所需的语义规则，本节介绍分析器并描述分析过程[52]。

## 4.3.3.1 分析器

语义分析器基于框架生成规则进行规则匹配和对象组合。框架生成规则描述了低一级语义对象组合生成高级语义对象的过程，这一过程类似于 CFG 所描述的由词组合成短语，再组合成高级短语，最后组合成句子的过程，但是没有句法规则所要求的词序和邻接限制。

由于没有词序和邻接限制，规则匹配采用模板匹配方式。每条规则定义的局部成分，对应有一个模板，模板的槽对应规则右端的语义项。每生成一个新的语义对象，考察所有模板，将其填入与之相符合的槽。如果模板中的槽都已被填入，即称模板被填满，规则得到匹配。为支持省略分析，规则中允许出现可省略的语义项（以下称省略项），则模板中除了省略项对应槽（以下称省略槽）之外的槽都被填入，模板也可得到匹配。显然，对于模板中槽的填入，当前句成分应该比上下文中成分优先，如果可以找到当前句成分填入，就不能认为出现了省略。为此，分析器采用双层多遍的分析流程，如图 4-7 所示。

在双层多遍的分析流程中，A 层进行当前句成分的匹配，B 层进行上下文中成分的匹配。首先进行 A 层循环，考察所有规则，进行模板填入，直到没有新对象生成。由于规则按照生成顺序由低到高排列，实际上只要对所有规则顺序进行一遍匹配，即可完成 A 层循环。之后进入 B 层循环，考察那些除省略槽外填满的模板，看能否通过省略分析，如果能通过，则在生成新成分后回到 A 层循环，如果不能通过，顺序考察其它规则。如果考察完所有规则都没有新成分生成，则整个语义分析过程结束。注意 B 层循环中只要生成一个新成分就要回到 A 层循环，考察这一新成分对于当前句成分的影响。

语义分析结束后，得到若干以框架形式表示的语义对象，其中对应于句子的问句框架即为分析结果，我们也称问句框架为意图框架。由于模板匹配过程未考虑规则间的互斥关系而产生的冗余，分析可能得到多个问句框架，我们通过启发式的打分方法选择最佳的一个。打分策略是当前句中词每个 10 分，上下文中词每个 2 分，分数最高的框架实际上就是匹配当前句中词最多，或者是当前句中词相同但引用上下文中词最多的那个对象。同样利用这一启发式打分策略，对于语音识别和句法分析得到的多候选，选择进行语义分析后分数最高的候选，从而实现消歧。

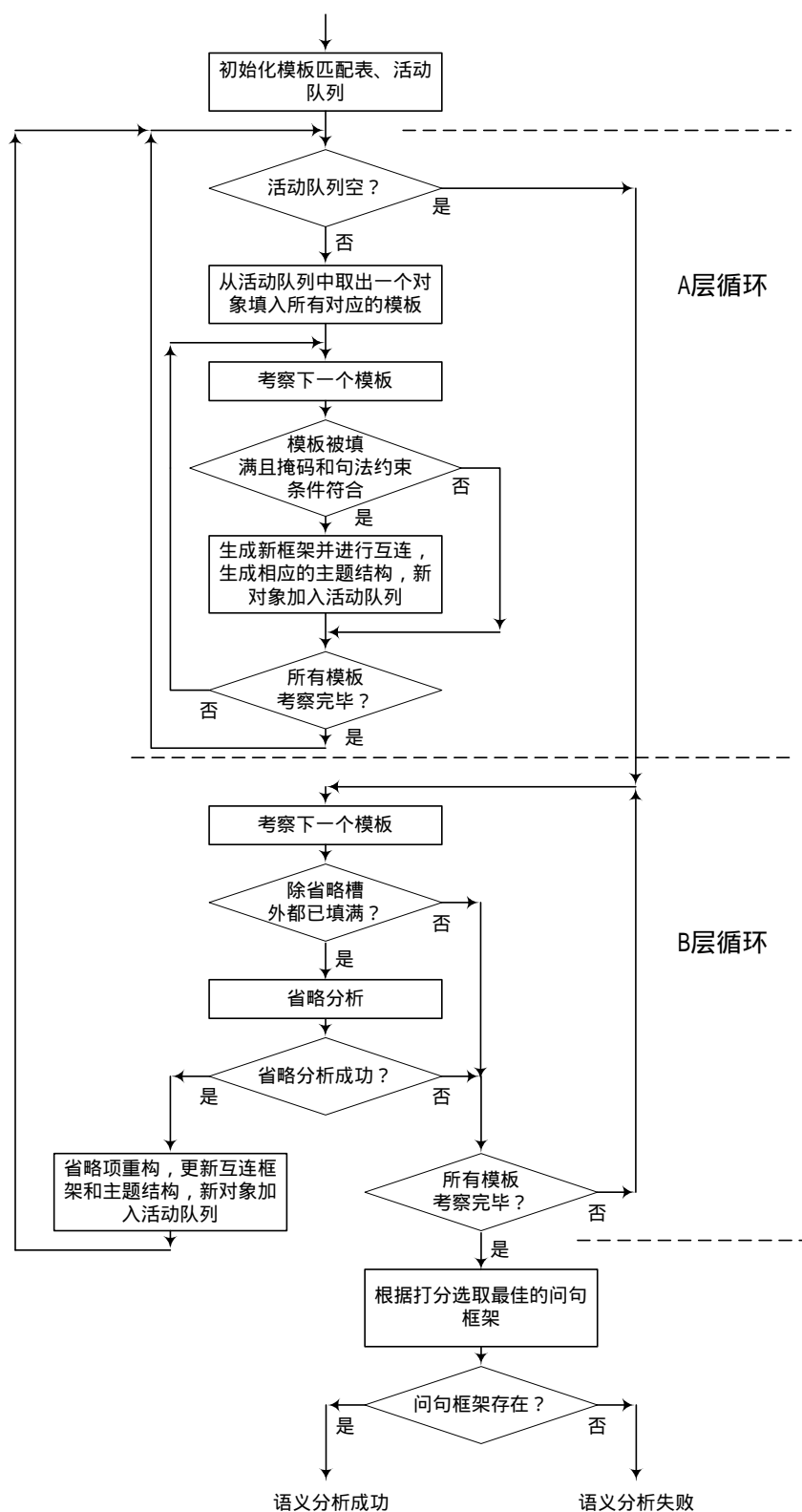


图 4-7：基于互连框架的语义分析方法流程

局部成分的模板匹配要避免出现重复匹配,即同一词被模板中多个槽使用。为此,我们为每个语义对象设置一个掩码(mask),掩码中每一位对应句子中的词。模板中各个槽必须保证掩码不冲突。

句法约束条件由语义规则定义,根据规则需要可繁可简。第二版考虑到语料中句子大多比较简单,没有使用象第一版那样复杂的句法约束条件,而是作了简化。一种简化是给规则设置一个公共句法标记,如从字短语要求右端的“从”和“地点”在同一个介词短语(PP)中。另一种简化是对部分规则加入词序和邻接限制。

#### 4.3.3.2 分析过程

本节通过实际句子的语义分析过程来说明上一节所给出的双层多遍分析流程,同时也介绍一下互连框架如何生成。

例(8)输入的时候,上下文中记录有“九号楼”的地点对象。

例(8) 附近哪个食堂比较便宜?

分析过程见图 4-8。

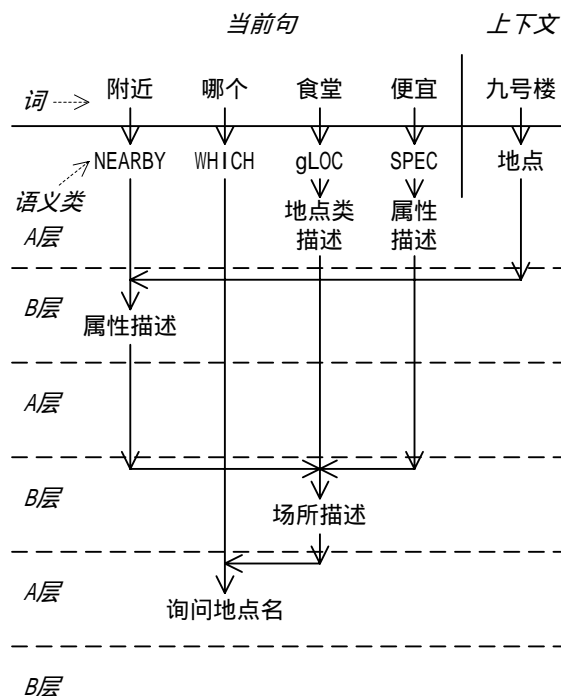


图 4-8：语义分析示例

图 4-9 给出最后所得到的互连框架分析结果。

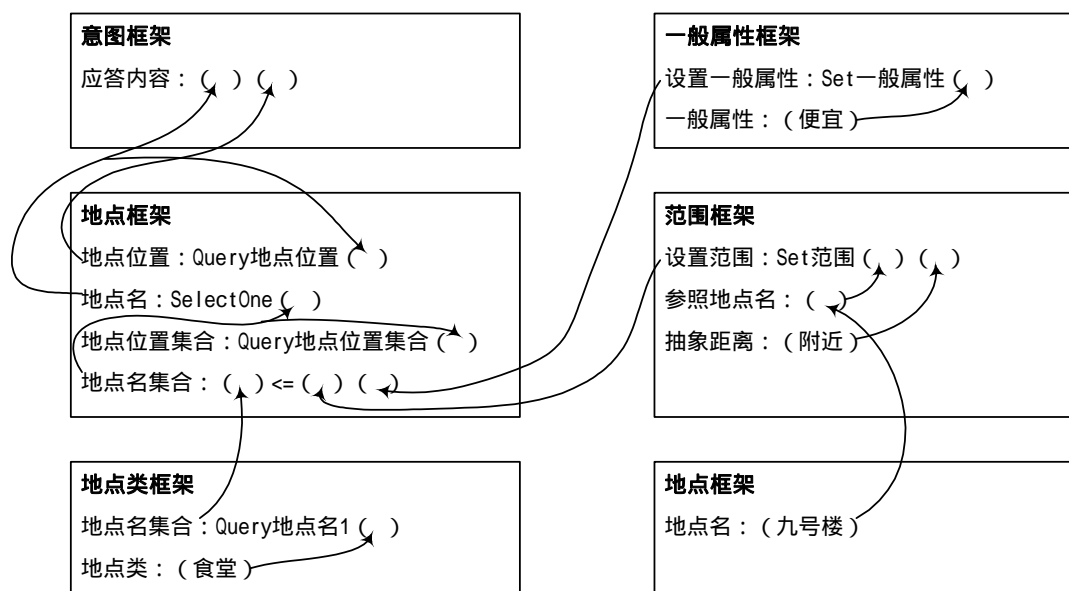


图 4-9：互连框架图示

语义分析过程实际上就是图 4-8 所示的语义对象组合的过程，对应于图 4-9 中框架互连的过程。

下面简单描述一下分析过程。例(8)作为当前句，句中的关键词对应于基本的语义要素，在 A 层循环中生成了地点类描述和属性描述两个对象，分别对应于地点类框架和一般属性框架。由于在不考虑上下文的前提下已经无法生成新的对象，转入 B 层循环，生成了对应于范围框架的属性描述对象。其中地点框架拷贝自上下文，与生成的范围框架进行互连。回到 A 层循环，依然无法生成新对象，转入 B 层循环，综合地点类描述和两个属性描述对象，组合生成对应地点框架的场所描述对象，组合过程中，地点类框架、一般属性框架、范围框架分别与生成的地点框架进行互连。回到 A 层循环，与疑问词组合生成一个问句对象，所对应的是意图框架。

意图框架也称问句框架或应答框架，其中应答内容槽中包含了两个应答入口，分别对应两条应答查询链。前面图 4-5 对应的便是其中对应于地点名的应答查询链。每个框架中包含了查询链片断，对应于语义信息的查询义，框架互连过程中查询链片断也连接起来，使整句的查询义得以动态生成。

语义分析过程同时也生成了主题结构，主题结构主要的生成方式是在语义规则匹配时触发关联规则，另外一个生成方式是在省略发现时触发省略项重构规则。例(8)生成的主题结构如图 4-10 所示。

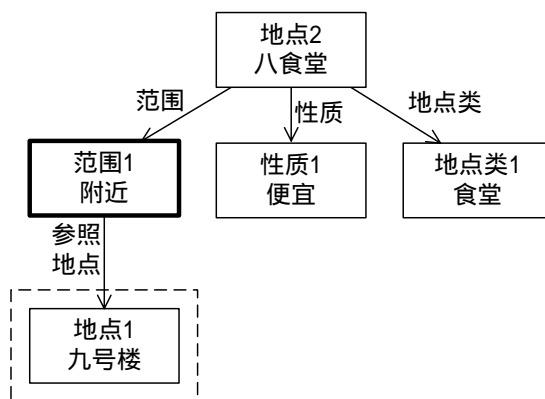


图 4-10：主题结构图示，虚线框指示该项目来自上下文

图 4-8 的分析过程第一次进入 B 层循环时，规则 12 发现省略，使用省略项重构规则生成一个地点项目，代替正常情况下由关联规则指示的地点项目与范围 1 项目相关联。之后省略分析模块以这个省略项为基准，与上下文中的主题项目进行比较，寻找符合匹配条件的引用项，得到上下文中的地点 1 项目。然后根据主题结构进行框架拷贝，将地点 1 项目对应的地点框架（即：九号楼）拷贝过来，并根据地点 1 项目所指示的框架槽，将该地点框架与规则 12 所生成的范围框架进行互连。

上下文信息与本句信息组合过程中对互连框架的槽间指针的调整，称之为指针重定位。第一重重定位是将当前框架中相应的远指针重定位到引用项对应的框架和槽上，即上述范围框架与所引用的地点框架间的指针重定位。第二重重定位发生在引用项是组合项目时，引用项及其子项目对应的一系列框架内的远指针都需要进行调整。

如上所述，主题结构生成与互连框架生成是同步的，其对语义分析起的作用是指导上下文信息对应框架的生成与互连。主题项目与框架间有一一对应的关系。上例中，生成一般属性框架和地点类框架时，也生成性质项目和地点类项目。引用地点项目时，拷贝地点框架。框架间互连的同时，项目间的关系也建立起来。

### 4.3.3.3 与句型模板方法的比较

我们对 EasyNav 系统先后使用的两种语义分析方法比较一下优缺点。

句型模板方法有若干优点：1) 列举句型，涉及因素较少，规则简单易写；2) 查询与应答策略直接与模板号相联系，可以解决言外之意的的问题；3) 与句法分析模块相对独立。但也有很大的缺点：1) 规则生成能力弱；2) 规则与程序相互不独立（依赖于模板号的定义），并且也不直观。

互连框架方法的优点：1) 规则生成能力强；2) 查询与应答策略显式地出现在规则中；3) 与句法分析模块相对独立。缺点则是规则复杂，除了基本的语义和句法以外，还涉及查询、应答、上下文等内容，书写困难，需要组合测试，并导致语义分析器代码比较复杂。

可以看到，互连框架方法针对句型模板方法的两个缺点作了改进，但随之而来的代价就是规则复杂度提高。

### 4.3.4 方法评价

从语义分析四个层面划分的角度来看，互连框架方法的分析层面比较高。意图框架中的意图信息对应于模式划分层面，框架互连的形式对应于结构表示层面，互连框架内包含的查询链对应于意义计算层面，辅助框架生成的主题结构则对应于上下文处理层面。同时，意义组合原则也贯穿于后三个层面之中。

从分析和表示复杂句子的角度来看，互连框架可以有效地表示复杂句子。动态生成的机制让分析器具有这样的能力：以较少的描述成分组合的规则来刻画自然语言的句子。这一机制能够有效描述递归结构的句子。

## 4.4 其它语义分析表示方法

作为对互连框架方法的比较，考察其它口语对话系统中所采用的语义分析表示方法，以进一步强调互连框架方法的特点。

## 4.4.1 MIT-SLS

MIT-SLS 的系列系统采用相似的技术，以航班信息查询系统 Mercury[64]为例介绍其使用的语义分析表示方法。语义表示主要被生成模块 GENESIS[65]所使用。生成模块包含查询（生成 SQL 语言）应答等多项生成功能。

系统中有两种形式的语义表示：语义框架和表单，分别对应于语言学的和非语言学的表示，如图 4-11 和图 4-12 所示。语义框架具有层次性，但由于未包含计算信息，这一层次性不能为生成模块所用。表单为非嵌套的单个语义框架，直接对应于任务知识，易于被后续模块处理。实际使用中，句法语义分析得到的语义框架被转化成为表单，以便生成模块处理。

```
Clause: EXIST
  Topic: FLIGHT
    Quantifier: INDEF
    Predicate: SOURCE
      Topic: CITY
        Name: Boston
      Predicate: DESTINATION
        Topic: CITY
          Name: San Francisco
        Predicate: TIME
          Topic: DATE
            Day: Friday
```

图 4-11：问句“Is there a flight from Boston to San Francisco Friday?”对应的语义框架

```
<flight_leg>
airline: "UA"
flight_number: 94
source: "BOS"
depart_time: "3:00 p.m."
```

图 4-12：一个表示航班的表单

可以看到，由于语义框架不包含计算信息，不得不将其转化成为表单进行处理，相当于退化到用任务知识来进行语义理解，从而削弱了语义表示的描述能力。

### 4.4.2 LIMSI

LIMSI 的系列系统使用格框架方法进行语义分析表示[3]。格用来描述句法成分间的语义关系，如动词的施事格、受事格、工具格等等。格框架即为与谓词相关的各个格构成的框架，或者进一步的，是与概念相关的各项约束属性构成的框架。格框架方法以框架为中心，考察是否符合语义类和句法约束条件，而不考虑整句的句法结构，分析过程是一个模板匹配过程。格框架相当于前述的表单。提取格框架中的信息依据预定义的格式生成查询和应答。

与以上口语对话系统中的语义表示相比，互连框架方法的分析层面较高。由于框架中包含了意义计算信息，层次结构表示的优点得以体现，查询和应答生成不再是依据任务知识预先定义，而是在语义分析过程中动态生成。

## 4.5 小结

本文提出的互连框架、查询链和主题结构都是依据意义组合原则生成，具有比较强的描述能力。一方面层层嵌套的形式可以描述复杂的语义，另一方面支持省略使得用户可以连续地提问。这些语义表示的另一特点是表示单位的自足性，表示和计算相一致。语义表示单位（即框架）内部提供信息的计算方法。将句子用互连框架表示出来，互连框架内含的应答查询链就已经提供了对整句意图的计算。这一自足性使得对上下文信息的引用简洁直接。

语义分析将自然语言句子转换成为计算机能够进行计算的语义表示。意义组合原则是通过自底向上的语义成分组合过程实现的。为实现更高层面与上下文信息的组合，采用双层多遍的分析流程，上下文分析在得到部分表示时即进行，语义分析和上下文分析紧密耦合。

互连框架方法的问题主要是规则复杂，需要组合测试。尤其要考虑上下文中对象和当前句中对象的组合，使得其与单纯句子语法系统的规则完全不同。

## 第五章 省略分析

### 5.1 引言

指代和省略是连贯的文本中必然出现的现象，是语言学家和计算语言学工作者所研究的重要课题。正确地理解指代和省略现象及其相互关系，对自然语言处理有重要的意义。本文重点研究汉语口语对话系统中出现的省略现象。一方面，汉语中的省略现象要比英语频繁，英语中使用指代的地方汉语中往往使用省略[67]；另一方面，口语对话中省略现象大大多于书面语[68]。这都表明汉语口语对话系统中的省略分析具有重要的意义。

指代和省略分析属于局部上下文的分析，在口语对话系统中向进行单句分析的句法语义分析提供上下文的语言信息。口语对话系统支持指代省略分析会使对话过程更加连贯自然，对话形式更加口语化，对话界面更加友好与智能，对整个系统性能的提高会有较大的贡献。

#### 5.1.1 指代

##### 5.1.1.1 指代现象

指代 ( anaphora ) 指句中短语与前文提到过的对象对应，前文中该对象对应的短语称为先行语 ( antecedent )。英文中指代的类型有：代词性指代、名词性指代、表层指代等[50][69]。下面是一些例子，大多数取自文[50]。

例(9) Jack lost his wallet in his car. He looked for it for several hours.

例(10) Jack forgot his wallet. Sam did too.

例(11) Jack lost the race. It surprised me.

例(12) You should come here.

例(13) Helen bought a car and a boat yesterday. She paid too much for the car.

例(14) I saw two bears in the woods. Bill saw some in the park.

例(15) Tell me John's grade in CSC271. Give me it in MTH444 as well.

例(16) Tell me John's grade in CSC271. Give me Mike's in MTH444 as well.

代词性指代有：代名词、代动词、代句子，例子分别见例(9)、例(10)、例(11)。代动词也可归入 VP 省略，正如例(10)中在用 did 指代前一个动作的同时，还省略了动作的参数，如例(10)中的受事。通常主要研究的是代名词，也就是通常所说的代词。代词指代的一个特殊情况是直指，即指云不是语句上下文中的对象，而是情境上下文中的对象，如例(12)。

名词短语有专名、不定指和定指。通常，专名直接对应指云，不定指描述对应一个尚未在语句上下文中出现的对象，定指描述指代语句上下文中的对象，只有定指描述才属于指代，如例(13)。

注意代词和定指指代范围上的差异，通常代词指代的先行语在局部上下文中，即上一句或者本句的其它子句中，而定指指代的先行语可以在全局上下文中更远的位置。

表层指代 ( surface anaphora )，也可称为意思指代、类别指代，指的是引用会话上下文中对象信息但指称的是新对象。如例(14)，并非对应同一指云，而是有相同的类别。如例(15)和例(16)，有相同或部分相同的描述信息。注意表层指代在本文中归为一种省略现象进行处理。

省略有时也会被归为指代的一种[69]，不过我们认为省略应该与指代分开来研究。

#### 5.1.1.2 指代分析

指代分析通常采取“筛选-倾向选择”两步骤的流程[70]。筛选指用约束条件匹配指代项和引用项。倾向选择指考察符合匹配条件的引用项间的区别。

会话理论的研究对指代分析有很大的促进作用，如会话表示理论[71]和中心理论[72]。

文[73]中给出了一个指代分析算法，用历史记录管理上下文。算法考虑了多种指代现象和约束条件，但对上下文管理和倾向选择问题都未作考虑，因而只是简单的示意算法。

比较有名的指代分析算法有：早期的 Hobbs 算法[74]、Lappin 和 Leass 提出

的算法[75]、Strube 基于中心理论的算法[76]。近期 Computational Linguistics 杂志所出的指代分析专辑[77]代表了目前指代分析研究的水平。这些算法的一般特点是避免利用过多的领域知识和语言知识，而更多利用大规模标注语料和表层语法知识如 POS 或是浅层分析结果，分析对象是不受限文本，正确率在 85%左右。

下面简单考察一下口语对话系统中进行指代分析的情况。

一个是地点信息查询系统中关于地点的指代，以 Voyager 系统[23]为例，通过保留当前位置和上一位置，实现诸如例(17)和例(18)中的指代分析。

例(17) What is their address?

例(18) How far is it from here?

一个是多模式对话系统中，指代要与其它模式如触摸相结合进行分析[78]。

### 5.1.2 省略

#### 5.1.2.1 省略现象

省略 (ellipsis) 通常定义为语法上不完整的结构。简单地说有名词、动词、形容词等的省略，具体研究则是按包含省略的语法现象来划分。

例(19) Holly Golightly won't eat rutabagas. I don't think Fred will, either.

例(20) John liked Mary, and so did Bill.

例(21) John liked the girls who Mary did.

例(22) John sent the flowers to Lucy before he did the chocolates.

例(23) Mary likes flowers, and Rosa [likes] chocolates.

例(24) John threw flowers at Mary, and [John threw] chocolates [at Mary] too.

例(25) Someone called, but I can't tell you who.

例(26) A car is parked on the lawn -- find out whose.

例(27) Rex wrote a more interesting abstract than [the abstract] I did/wrote.

例(28) Rex wrote a more interesting abstract than [the abstract] I thought [he did/wrote].

动词短语省略 (VP Ellipsis, VPE) 指谓词缺失, 并可在上一句中找到先行语[79], 典型的句间 VPE 如例(19)和例(20)[80]。VPE 是英语中研究最多的一种省略现象, 下述的几种省略也归属于 VPE[80]。先行语包含省略 (Antecedent Contained Ellipsis/Deletion, ACE/ACD) 指在关系从句中出现的 VPE, 如例(21)。伪间断 (pseudo-gapping) 和间断 (gapping) 省略指主语宾语保留而动词省略, 前者使用代动词, 如例(22)[81], 后者不使用, 如例(23)。剥除 (stripping, or bare ellipsis) 省略, 指只出现单个名词短语的 VPE, 如例(24)。

其它研究较多的省略现象还有: 冲刷 (Sluicing) 省略指句子或子句只有单独疑问词[82], 如例(25)和例(26)。比较句型省略 (Comparatives Ellipsis, CE) 指由 than 句型引发的省略[83], 如例(27)和例(28)。

零代词指原本可以用代词的位置出现省略, 是英语以外的语言中常常出现的一种省略现象, 如日语[84]、西班牙语[85]等。

口语对话过程中简短应答或提问时常出现片段型省略[86], 如例(29)和例(30)。

例(29) A: Who saw Mary? B: John.

例(30) A: A student saw John. B: Who?

汉语中对省略现象的研究较少[87], 主要从语文学角度研究, 受文体限制。

### 5.1.2.2 省略分析

英文中省略分析通常以 VPE 为研究对象。要解决三个问题[79]: 1) VPE 在语法上是否被允许 (licensing); 2) VPE 与先行语间有何结构联系; 3) 省略信息如何重构。具体分析依赖于具体的省略现象及相应的语法结构。

省略信息重构通常以逻辑语义学作为分析工具[88]。简单地说, 是对前一句进行  $\lambda$  抽象提取部分信息, 再用当前句信息进行  $\lambda$  还原, 实现省略信息的重构。量词、辖域、指代约束等因素会令  $\lambda$  抽象出现歧义, 增加问题复杂性。例(31)是个简单的示例:

例(31) Dan likes golf, and George does too.

前一句的逻辑式为：like(dan,golf)

进行 $\lambda$ 抽象后得到： $\lambda x.like(x,golf)$

$\lambda$ 还原得到当前句逻辑式： $[\lambda x.like(x,golf)](george)=like(george,golf)$

下面考察一下口语对话系统中进行省略分析的情况。

以LIMSI的RailTel为例,系统的语义表示把多回合提供的信息融合在一起,填入格框架[15]。可以认为当前句省略了之前已经提供的信息。用户的应答可以是片断,根据提问句子可以恢复例(32b)。要考虑缺省值,如例(33)中月份的缺省。

例(32a) System: What is your departure city?

例(32b) User: [My departure city is] Paris.

例(33) I would like to go on the 6th [of the current/next month].

### 5.1.3 会话理论

指代省略作为一种会话现象,其计算受到会话性质的影响。会话理论如会话表示理论[71]、中心理论[72]的提出,推动了指代省略分析的发展。

#### 5.1.3.1 会话

会话(discourse)指一组句子,或者说篇章。会话、语用、上下文(或称语境)间有很强的联系,常常相互替用。语用考察语言与使用语言的上下文之间的关系,使用上下文包括会话上下文(语句上下文、双方目的、双方认识和假设)、情境上下文(说话人、地点、时间等)、世界知识、文化社会知识等[49][89]。其中语句上下文指的是与当前句相关的句子序列,又分全局上下文和局部上下文两个考察方向。全局上下文指的是整个会话的结构,考察当前句在会话的层次结构中所处的位置。局部上下文指的是前后的句子,考察当前句与相邻句子间的联系。

衔接(cohesion)和连贯(coherence)是会话的特性,用以刻画会话中句子间的联系。前者指词汇、指代省略的一致以及连词的连接,即表层结构的一致

性；后者指句子间语义上的关联，如隐含的因果[90][49]。会话连贯性分全局连贯性和局部连贯性两个考察方向[72]，分别对应于前述的全局上下文和局部上下文。会话的含义超出各部分含义的总和，这一超出的部分来自为建立连贯性而隐含的含义[91]。

### 5.1.3.2 会话结构

会话结构是会话含义计算的基础。Grosz 在 1983 年提出了一个会话模型，将会话结构分离成三个相互作用的部分：篇章结构（句子序列结构或语言结构）、意图结构、关注状态[92]。

篇章结构即前述的会话分段[50]及段间的嵌套关系。会话段（discourse segment, DS）对语言理解有影响，比如说指代在段内和段间处理方式不同。

意图结构由整个会话的会话意图（discourse purpose, DP）和各个会话段的会话段意图（discourse segment purpose, DSP）构成，会话意图和会话段意图与 Grice 定义的句子意图有类似的特性，即要通过识别性来定义。DSP 之间的支配关系构成一个支配层次结构。

关注状态（attentional state）包含在某一点上最突出（salient）的对象、属性、关系和意图。关注状态是会话参与者关注焦点的抽象，它总括了之前的篇章中对当前句处理很重要的信息，因而避免了保留所有句子的完整纪录。文[92]中使用焦点结构来刻画关注状态。每个会话段对应有一个焦点空间（focus space, FS），焦点结构就是焦点空间的集合。焦点结构对指代的可引用区域、突出度计算提供了会话结构上的依据。

需要特别指出的是会话模型的动态性，无论是篇章结构、意图结构还是关注状态，都不是先验的，而是会话过程中实时构建的，随着会话进行而不断充实发展。

### 5.1.3.3 会话关系

会话关系，指的是依据会话连贯性定义的句子之间的关系 [50]。Hobbs 认为不需要先进行会话分段以确定会话结构，在结合会话关系（Hobbs 称之为连贯关系）进行领域推理之后自然可以得到会话结构[93]。

文[93]提出 5 种会话关系，分别为：结果 Result，解释 Explanation，并列

Parallel, 拆细 Elaboration, 场合 Occasion。文[91]抽象成 3 种会话关系：相似 Resemblance, 因果 Cause-Effect, 接近 Contiguity。会话关系对应于语义连贯性, 因而其定义主要考察句子对应的命题间的关系。

会话关系作为语义连贯性的描述, 对命题提供了约束。运用这些约束我们可以进行推理计算[49]。诱导 (abduction) [94]推理在计算会话关系的同时可以找到代词所指。在因果关系的会话中, 代词所指受语义连贯影响要比受表层结构影响大, 如例(34), 两句话的表层结构完全相同, 代词所指却不同。动词语义不同, 建立的因果链不同, 对代词变元的赋值也不同。

例(34) The city council denied the demonstrators a permit because

a. they feared violence. [they=the city council]

b. they advocated violence. [they=the demonstrators]

Kehler 考察会话关系与指代分析间的关联, 试图把会话关系、结构与指代统合成一个整体[91]

#### 5.1.3.4 中心理论

连贯性不仅有如何进行连贯的问题 (即会话关系), 还有个连贯性强弱的问题。Grosz 在基于焦点的会话结构理论[92]的基础上提出中心理论[72], 用以对会话结构中局部的关注状态进一步建模。Grosz 认为, 局部连贯性强弱, 表现在指称表达式的选择及不同指称表达式所对应的推理代价, 而这可以用中心理论来计算。

中心 (center) 的概念来自于焦点。会话段 DS 中每句话 U, 都对应一个前向中心集合  $C_f(U, DS)$ , 除了会话段首句外每句话都有且仅有一个后向中心  $C_b(U, DS)$ , 可以简写成  $C_f(U)$  和  $C_b(U)$ 。前向中心集合中的元素根据在句中的显著度 (prominence) 有级别 (rank) 顺序, 排序受语法位置、词序、子句、词义等因素影响。第  $n+1$  句  $U_{n+1}$  的后向中心对应第  $n$  句  $U_n$  的一个前向中心。如下述定义所示:

- 在  $U_n$  中:  $C_b(U_n)=a$ ,  $C_f(U_n)=(e_1, e_2, \dots, e_p)$ ,  $a=e_k$ ,  $k$  为某一值。
- 在  $U_{n+1}$  中:  $C_b(U_{n+1})$  实现  $e_m$ , 且对任意  $j < m$ ,  $e_j$  不在  $U_{n+1}$  中实现, 即,  $e_m$  在  $U_{n+1}$  中实现, 且没有更高级别的  $e_j$  在  $U_{n+1}$  中实现。

定义句子间的三种变化关系:

1. 中心延续： $C_b(U_{n+1}) = C_b(U_n)$ ，并且该实体为  $C_f(U_{n+1})$  中级别最高的元素。
2. 中心维持： $C_b(U_{n+1}) = C_b(U_n)$ ，但是该实体并非  $C_f(U_{n+1})$  中级别最高元素。
3. 中心转移： $C_b(U_{n+1}) \neq C_b(U_n)$ 。

中心理论提出中心实现的两条限制规则：

1. 如果有  $C_f(U_n)$  中的元素在  $U_{n+1}$  中由代词实现，则  $C_b(U_{n+1})$  也必须由代词实现。换言之，除非  $C_b(U_{n+1})$  由代词实现，不然句中不能有任何元素由代词实现。
2. 句子间的中心延续关系优先于中心维持关系，而中心维持关系又优先于中心转移关系。

下面我们用例子解释一下中心理论但是，

例(35) John hit Bill. Mary told him to go home. [ him = John ]

例(36) Bill was hit by John. Mary told him to go home. [ him = Bill ]

在例(35)和例(36)中，前半部分的句子表达意思是相同的，但一个是主动句，一个是被动句。根据语法位置的级别次序，主语 宾语 其它，例(35)中级别最高的  $C_f$  是 John，例(36)中级别最高的  $C_f$  是 Bill。进而根据规则 2，中心延续优先，后一句的  $C_b$  应为前一句中级别最高的  $C_f$ ，而根据规则 1，句中唯一的代词必须实现  $C_b$ ，则 him 所指在对 John 和 Bill 两个候选进行选择时，应该选取相应的  $C_b$ ，例(35)中为 John，例(36)中为 Bill。

## 5.2 语境省略

不同于上一节省略分析综述，本节主要从本系统的背景，即汉语口语角度，考察省略现象，并给出我们对于省略的定义、分类、研究策略，作出自己的定位。

### 5.2.1 汉语和英语中省略的不同

汉语和英语中的省略现象有很大不同，吕叔湘在[95]中举了若干例子说明。下文中主要以这些例子来加以比较。

文[95]中指出，往往一个句子里有一个成分，外语必须说出来，汉语可以不

说，甚至必须不说。

例(37) 他问过许多人，[他们]都不知道。He has asked many people. *Nobody* knows.

例(38) 她有一个儿子，[他]去年参的军。She has a son, *who* joined the army last year.

例(39) 这本书我用不着了，你可以[把它]拿去。I have no more use for this book. You may have *it*.

例(40) 这是一本好词典。我已经买了[一本]，你要不要买[一本]？This is a good dictionary. I have got *one*. Don't you want to have *one*?

例(41) 他问我[ ]去不去。He asked me whether *I* would be there.

例(42) 赤膊磕头之后，居然还剩几文，他也不再赎毡帽，统统喝了酒了。  
(鲁迅：阿 Q 正传) After kowtowing with bare back he still had a few cash left, but instead of using *these* to redeem his felt hat from the bailiff, he spent *them* all on drink. (杨宪益、戴乃迭译本)

例(43) 你不去我不去。I won't go *if* you are not going.

例(44) 长江黄河。The Yangtze *and* (~or) the Yellow River.

例(45) 他晚上工作，白天睡觉。He works *by* night and sleeps *by* day.

例(46) 我睡行军床。I'll sleep *in* the camp bed.

前四个例子是典型的英语中使用指代，汉语中可以使用指代，但更多的是使用省略的例子。这些例子中，尽管可以认为是动词短语中的施事或者受事省略，但是并没有英语 VP 省略所要求的句子间并列结构，从其性质看，这些省略更接近于英语中的代词。

例(42)中，原文对“几文”跟“赎毡帽”、“喝酒”的关系没有明文交代，汉语中是允许的，但英语中不行，因此译文中必须增添必要的词语，其中指代词 *these* 和 *them* 都指 cash。尤其值得注意的是，这个例子中的省略并非从语法上考察，而是语义信息上的不完全。句子中没有明确指出部分语义对象之间的关系，“几文”应该作为“赎”和“喝酒”动作表代价的格。

例(43)和例(44)是省去连词的例子，例(45)和例(46)是省去介词的例子。不过这种省略更多的是语法的一部分，表明汉语语法很灵活，也完全可以通过语法

来解决，而不需要考虑上下文。

文[95]中还指出，指代词、连词、介词，英语中有时候也会省掉，但是英语中最容易省掉的是动词或者复合动词的一部分，这是汉语中决不允许的。例如英语中可以说：“My sister works in a factory, and my brother on a farm.” 汉语中必须说：“我姐姐在工厂工作，我哥哥在农场工作。”

### 5.2.2 语境省略定义

牛津英语语法[96]中为省略作了一个定义：省略是句子中不存在，但人可以恢复的语法成分。这一定义是一个一般性的定义，不论对于英语还是汉语都适用。但是定义中没有回答如何计算的问题：如何发现句子中不存在的成分？又如何恢复该成分？

文[50]中给出了一个计算性定义：省略是不完整的语法结构，通常是一个具有不完整句法结构的子句利用之前的完整子句进行重构。典型的例子是并列结构中的 VP 省略、并列句子中的 VP 省略和连续询问中的片断型省略，如例(47)、例(48)和例(49c)。

例(47) Helen saw the movie and Mary did too.

例(48) Some think that Jack will win the race next week. But he never will.

例(49a) A: Did Sam find the bananas?

例(49b) B: Yes.

例(49c) A: The peach?

这一定义的要点是：1) 以句法完整性作为省略检测的依据；2) 以前一子句为基准进行省略重构。

在对汉语省略现象研究过程中，我们认为这一定义具有局限性。我们从如何检测省略的角度，对省略现象作一计算上的划分。通过判断句法结构是否完整，来确定是否是省略的，划分为**句法省略**；通过上下文来判断是否是省略的，划分为**语境省略**。文[50]中所定义的是句法省略，而我们着重研究的是口语对话系统中出现较多的语境省略，如例(50)-例(53)。

例(50a) 用户：请问计算机系学生住在哪里？

例(50b) 系统：九号楼。

例(51a) 用户: [九号楼]附近都有哪些食堂?

例(51b) 系统: 七、八、九食堂。

例(52a) 用户: [九号楼附近]哪个食堂的东西比较好吃?

例(52b) 系统: 九食堂。

例(53a) 用户: 那[九号楼附近]比较便宜的[食堂][是哪个]呢?

例(53b) 系统: 八食堂。

根据我们对汉语省略现象的初步研究,我们对语境省略也进行一个计算上的定义:语境省略是语义结构中可以补充的信息,且上下文允许并提供这些信息。这一定义中有如下要点:

1) 语义而非句法。我们从前面对汉语和英语中省略的比较可以看到,汉语中省略的发生并不需要并列结构或者片断这么强的句法条件,例(37)-例(40)中只是省略了代词,例(41)中代词的省略甚至是语法的需要,例(42)中的省略更不能说是句法结构上的不完整,而只是动词的代价格需要填入恰当的信息。

2) 可补充而非不完整。例(42)中的省略信息是动词的代价格需要补充信息,但是如果没有这项信息,“赎毡帽”对应的语义结构依然是完整的。例(52)中的省略信息是例(51)中延续下来的查询约束条件,而在没有例(51)的上下文的情况下,例(52)本身的语义结构也是完整的。

3) 上下文允许。省略作为局部上下文现象,对其的判定需要考察局部连贯性。当这种连贯性很弱时,尽管当前语义结构可以补充,依然可能不从上下文中补充这一信息。5.3.3.3 节中会进一步讨论这一问题。

4) 上下文提供。上下文包括语句上下文和情境上下文。情境上下文所提供的信息主要指缺省值,5.3.4.2 节中会进一步讨论。对于句法省略,由于句间有结构上的依赖性,可以以上下文的语义结构为基础,由当前句提供信息,如5.1.2.2 节中所介绍的,先将前一句得到的逻辑式进行 $\lambda$ 抽象,再将当前句中的语义信息代入。语境省略由于句间不存在这么强的关联,因此要以当前句为基础,由上下文提供信息。

在这一定义中,省略检测和省略重构的原则作了改变。省略检测依赖语义结构知识而非句法知识,省略重构也立足于当前句而非前一句进行。

本文中不研究句法省略。句内省略对语法的依赖性更强,也可以看成是句

法省略，本文也不研究。下文中提到的省略一般都指的是语境省略。

### 5.3 省略分析

基于前述语境省略定义中的原则，提出了进行省略分析的主题结构方法。方法重点在于：一、分析了省略现象与指代现象的不同与联系，从而将指代分析处理方法延伸到省略分析；二、加强了语义知识的作用，并引入了场景知识和会话知识；三、动态生成反映关注状态的主题结构，适合用户主导对话系统的主题变化建模；四、采取统一的分析方法，对于各种省略成分处理方法一致，将片断型省略、缺省型省略也统一在同一个处理框架中。

#### 5.3.1 对于主题结构的讨论

在 4.3.1.3 节介绍了主题结构，作为承载省略分析的语义表示，本节要对主题结构对于省略分析的作用和意义作一讨论。

##### 5.3.1.1 主题变化

如 2.3.1 节中所讨论的，EasyNav 采用的是单回合交互模式。一个回合就是一次交互，主题切换频繁。尤其复杂的情况是同一主题类型，但不同主题的情况。这时仅仅认为段内主题相同、段间主题变化是不够的，还需要对段与段之间主题变化的程度进行建模。

例(54) 用户：三十号楼附近有哪些食堂比较好？

例(55a) 用户：一号楼附近有哪些食堂？

例(54) 用户：三十号楼附近有哪些食堂比较好？

例(55b) 用户：哪些食堂比较好？

例(54) 用户：三十号楼附近有哪些食堂比较好？

例(55c) 用户：一号楼附近有哪些[食堂][比较好]呢？

例(54) 用户：三十号楼附近有哪些食堂比较好？

例(55d) 用户：[三十号楼附近]哪些食堂比较便宜？

从上面例句看，同样是根据约束条件查询食堂，查询主题间却具有不同的变化程度，感觉上例(55a)和例(55b)是个新的查询主题，例(55c)延续性强，变化不大，例(55d)则介乎两者之间。对于主题变化，我们认为很难根据查询约束条件的变化，就断定主题是否有了改变。需要总结规则，进行更精细的建模。

由于主题的变化不能简单认为是从某一类型变化到另一类型，为刻画主题变化程度，需要一个结构来表示主题，通过对变化前后的结构进行比较得到主题变化情况，这个结构就是我们所提出的主题结构。

EasyNav 场景文本的特点是由用户主导的连续查询问句，这与通常 NLP 中考察的叙述型文本有很大区别。用户提出一个问句，相对应的就有一个查询表达式。当用户紧接着又提出一个问句时，可能是延续上一个查询，只不过补充了查询条件，也有可能是提出了新的查询表达式，但可能有些信息是从上个查询表达式中保留下来的。我们可以把一个查询表达式看作是一个主题，则新的问句可以对应于主题的延续或者切换，所要考察的是主题结构中语义信息的生成与维护。省略现象很大程度上表现在主题结构的渐进生成上，而局部上下文的延续性也反映在主题结构是否保持不变上，因而可以用基于主题结构的方法来处理省略分析。

#### 5.3.1.2 关注状态

尽管主题代表了交互过程中的用户意图，我们使用主题结构所表示的是关注状态，而非意图结构，参见 5.1.3.2 节。

文[92]中用焦点空间刻画关注状态。文[72]中进一步提出中心理论用中心刻画局部关注状态，用以指导代词指代分析。

我们提出的主题结构也是用来刻画局部关注状态的。由于代词指代和省略具有很多相似性，相应的，中心和主题结构也有很多共性。中心既是会话结构，也是语义对象，而不是一个句法概念。相对应地，主题结构包含语义对象及其相互关系，并且代表了对话的主题，在其上进行的指代匹配和引用项拷贝也不考虑句法结构。中心理论中的后向中心  $C_b$  只从上一句的前向中心  $C_f$  中寻找，具有局部特性。本方法中的指代匹配也只从上一句的主题结构中匹配。

中心理论主要有三个要点：1) 认为有些成分要比其它成分有更高的显著度，

并使用语法地位、词序、子句、词义等因素来对成分排序；2）代词与中心有很强的关联；3）会话的局部连贯性用中心的变化程度来刻画。

我们所提出的主题结构方法借鉴了中心理论的后两个要点。我们使用主题变化程度来刻画连续查询问句序列的局部连贯性，而用于表示主题的主题结构又与省略有很强的关联。

中心理论作为一般性的理论，较少利用世界知识。主题结构方法由于是以具体领域下的对话系统为背景，可以使用场景知识来帮助进行上下文建模。

### 5.3.2 省略分析处理步骤

语言学家研究省略现象时通常面向某一类现象，其分析方法通常依赖于该现象对应的语法特性，这种基于句法省略的处理策略不适用于 EasyNav 场景中的语境省略现象。需要针对语境省略的特点，提出自己的处理策略。

指代与省略都是局部上下文现象，具有相似性，我们对省略分析的处理策略借鉴了指代分析的处理步骤。指代分析通常采用筛选-倾向选择的步骤，其中，筛选步骤将当前处理的指代或省略项与上下文中的语义对象进行匹配，将不符合约束条件的对象滤掉，这个过程我们也称之为指代匹配。如果有多个满足条件的对象，则需要倾向选择步骤，根据某种量度挑选出一个最合适的，作为对应的引用项。这两个步骤对于省略分析也适用。

省略与指代的最大不同在于省略没有显式的语言标示。指代通常都有代词或者名词短语指示其存在，而省略在表层结构中没有相对应的成分。

这一不同导致省略分析需要增加两个步骤：省略发现和省略校验。省略发现根据语义知识列举可补充的语义信息，作为可能的省略。而这一可能的省略还需要根据上下文知识进行校验，即便上下文中能提供合适的语义信息，但根据主题变化情况依然可能判断省略不成立。这两个步骤回答了 5.2.2 节省略定义中的第一个计算问题：如何发现句子中不存在的成分？

省略发现是以语义分析为基础进行的。由于采取的是立足于当前句的省略重构原则，处理策略是由当前句的语义要素组合生成语义结构，在生成过程中如果有需要，才由上下文提供信息补充进语义结构中。换言之，在语义分析过程中，发现省略，即调用省略分析模块。省略发现实际上既是省略分析的第一个步骤，也是语义分析和省略分析连接的桥梁。整个省略分析处理步骤及与语

义分析间的流程关系如图 5-1 所示。下一节中将对省略分析各个步骤进行详细说明。

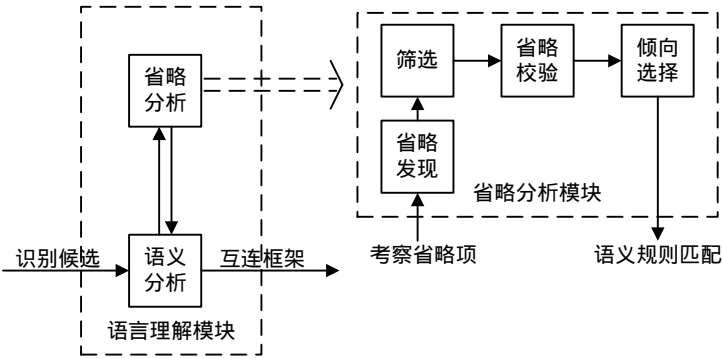


图 5-1：省略分析处理步骤

5.3.3 基于主题结构的分析

在上节介绍了“省略发现-筛选-省略校验-倾向选择”的处理流程后，本节具体介绍流程中各步的设计和实现。

5.3.3.1 省略发现

我们称框架生成规则右端有省略项的语义规则，为带省略项的规则。在语义分析的 B 层循环中，考察所有带省略项的规则，如果规则右端除了省略项以外的语义项都得到匹配，则认为这条规则可能得到匹配，也即发现了省略。如规则 12：

// 规则 12  
属性描述->地点(.) + 附近(NEARBY)  
省略：地点 ~地点 >> 范围 ^参照地点 =

规则 12 中地点项标示为省略项。如果语义分析过程中，找到了“附近”，但没有找到地点项，同时又从上下文中能找到地点项，此时认为发现了地点项的省略并触发省略分析，规则 12 仍然能够得到匹配而生成属性描述对象。

规则中语义项标注“.”的指示可省项，标注“\*”的指示可选项，虽然都

是指示语义结构中可以补充的信息，但可省项表示该信息是语义结构所必须的，可选项则表示该信息不是必须的。这一区别在寻找引用项失败后用以指示不同的处理。

为进行进一步的处理，要根据省略项重构操作构造一个省略项目，并添加到主题结构中。对规则 12 而言，省略发现之后，构造一个地点项目，该项目以关系“参照地点”关联到范围项目上。这一地点项目实际上是个空项目，因为没有对它赋值，没有将其与具体的框架和槽关联。构造省略项的作用是指示该省略对象的类别及其与其它对象间的关系，以便下一步筛选步骤中利用。只有在省略分析成功后，省略项才会被赋值，将该项目与重构的互连框架中的槽相关联。

### 5.3.3.2 筛选

筛选过程将省略项重构操作生成的省略项，与上下文中的主题项目进行比较，寻找匹配项。匹配原则是二者的语义性质的相似性。指代分析中筛选过程还要考察更多语法上的匹配性，如性别、数等等，目前的省略类型中都不涉及。

匹配条件因引用项是对象或者是属性而不同。如果引用项是对象，省略项与引用项的相似性用类别是否相同来判断。如果引用项是属性，省略项与引用项的相似性用该项目与其父项目间的关系是否相同来判定。

之所以不用类别相同作为属性项目的匹配条件，一个原因是有些项目具有不同类别，但所起作用相同的。如地点类和功能这两个主题类别，作为地点项目的属性时，主题关系都是地点类。描述地点类别，可以使用类别词，如“食堂”，也可以用一个功能描述，如“吃饭的地方”。从某种角度来说，是因为同一语义对象既可用原子项目描述，也可用组合项目描述。

另一原因是有些项目具有相同类别，但所起作用不同。如从到句中的源地点和目的地点，都是地点类别，但与路线父项目的关系不同，匹配时要以后者为基准。

### 5.3.3.3 省略校验

由于省略发现只是由于语义结构可以补充而发现了一个可能的省略成分，还需要省略校验最终判定这是否是一个省略。这一步骤显示了指代分析和省略

分析的一个差别：指代分析要从筛选成功后得到的引用项中选一个，而省略分析在筛选过程成功后可能一个引用项也不要。

省略校验利用会话知识来判定省略是否成立。这一处理的是比较微妙的主题类型相同但主题可能不同的情况，根据会话的连贯性强弱程度来判断是否有省略，而这一连贯性强弱的判断是依据场景知识作出的。

我们仿照中心理论中用中心延续、维持和转移来刻画局部连贯性的思想，提出主题延续、维持和转移来描述 EasyNav 场景中连续查询的局部连贯性。连贯性越强，允许省略的成分就越多。

首先我们给出根据场景知识判断主题变化的规则：

1. 如果父项目类别不符，或者与父项目关系不符，主题转移；
2. 如果父项目为组合项目，主题延续；
3. 如果是疑问词省略或是关键属性省略，主题延续；
4. 如果约束条件集合是空集，或者是上一句的子集，主题转移；
5. 如果以上条件都不满足，主题维持。

主题类型相同但主题不同的例子是若干约束条件下的地点查询。我们定义维持优先级为当前句中非重复出现（替换或是新出现）的属性中最高的优先级。我们用允许属性延续这个术语表示允许省略项作为属性去与上下文进行匹配，或者简单地说，承认该属性可以省略。下面给出依据主题变化来判断的属性延续条件：

1. 如果主题延续，所有的属性都允许属性延续；
2. 如果主题维持，比维持优先级高的属性允许属性延续；
3. 如果主题转移，不允许属性延续；但是依然允许作为对象引用。

主题变化建模的应用例子将在后面讨论具体省略类型时给出。5.3.4.1 节讨论的多约束条件查询是典型的例子。5.3.4.5 节讨论从到句时则会给出一个主题转移但允许作为对象引用的例子。

### 5.3.3.4 倾向选择

经过筛选和省略校验步骤，可能会得到多个符合条件的引用项，倾向选择

要依据某种判据选出一个。这种选择不象筛选过程那样严格，通常保留一个相对较好，在上下文中更为显著的。

代词指代分析过程中，倾向选择有较强的限制，如中心理论中要求代词优先由后向中心实现。在 EasyNav 系统的省略分析中，多项符合的情况并不多，因而只给出了简单的判据：对象优先于属性；没有对象只有多个属性则挑第一个。

### 5.3.3.5 疑问词省略

疑问词省略是比较特殊的省略类型，因而解决机制与其它省略不同。一句中只有一个疑问词，因此疑问词单独保存，而不在主题结构中表示。分析时直接将发现的省略疑问词与上下文中的疑问词比较，相同即分析成功。

### 5.3.3.6 寻找引用项失败的处理

如果经过省略分析的各步骤，从语句的上下文中寻找引用项失败，则有几种可能：1) 这是省略，省略信息来自情境上下文；2) 这不是省略，但语义结构依然完整；3) 这不是省略，语义结构因此不完整。具体是哪种情况，依赖于省略项的性质。

首先要考察是可选项还是可省项。可省项对语义结构必不可少，寻找引用项失败会导致语义结构不完整。可选项则可有可无，寻找引用项失败语义结构依然完整。

其次要考察省略项是值型省略还是函数型省略。这通过在省略项重构操作中标注“=”进行指示。值型省略在寻找引用项失败后根据场景知识为省略项设置一个缺省值。函数型省略则在寻找引用项失败后认为省略分析失败，语义规则得不到匹配。

### 5.3.3.7 重构过程

省略发现时的省略项重构操作，只是标示了省略项的语义性质。在确定了引用项后，还要有一个语义结构的重构过程。

首先是主题结构的重构。将上下文中的引用项拷贝到当前主题结构中，替

换原先的省略项。如果引用项是组合项目，要连同其子项目一同拷贝。

其次是互连框架的重构。将引用项指向的框架（以下简称引用框架）拷贝到当前互连框架中，并将与该框架具有互连关系的所有框架一同拷贝过来。注意由于查询完成之后有个更新过程，将查询链替换为查询结果，因而与引用框架有互连关系的框架，通常只包括对应组合项目子项目的框架。

最后是建立关联。引用框架要与当前互连框架中的框架互联。随着引用框架拷贝过来的框架间的远指针要重定位。这些框架还要与引用项的子项目建立关联。

重构中的一大问题就是上下文信息和本句信息的组合。框架是依据语义规则生成的，框架间的互连体现了语义信息的组合。但在设计规则中往往不会考虑两条本来无关的规则会因为上下文分析的缘故发生关系，相应地导致框架间互连出现问题。

举例来说，现有两条规则：

A → B C

D → E C

对象 A 由 B 和 C 生成，和 D 毫无关系。但在当前句中如果 C 省略，从上文中找到了 D，D 中包含了信息 C，则重构的结果相当于增加了规则：

A → B D

如果 A 中对于信息 C 的定义和 D 中的不同，互连就会出现问題。

具体的例子就是“地点类”，既会对应值型槽，又会对应函数槽，如果重构过程需要互连到一个函数槽，但引用框架中却是值型槽，就会出现问題。为此相应项目赋值时要求双槽关联，项目既指向一个值型槽，又指向一个函数槽，重构时按需要进行关联。相应框架中也需要增加信息，虽然这一信息在只考虑语义直接组合时用不到。这表明，为支持上下文信息和当前句信息的语义组合，要求语义表示的单位具有自足性，包含有足够用于计算的信息。

#### 5.3.3.8 更新过程

更新过程是在查询完成之后、保存上下文之前进行的，目的是将互连框架中的查询链替换成为确定的查询结果。在用户提问时，语义对象只是对应于一

个描述，但在系统应答后，该语义对象有了确定的指云。下文中如果引用这个语义对象，所指的是具体的指云，而非描述。更新也是为了保证指云的唯一性，因为查询函数如 SelectOne，返回值不是固定的。

### 5.3.3.9 细流程

综合各步骤，可以把图 5-1 的粗流程扩展成为完整算法，如图 5-2：

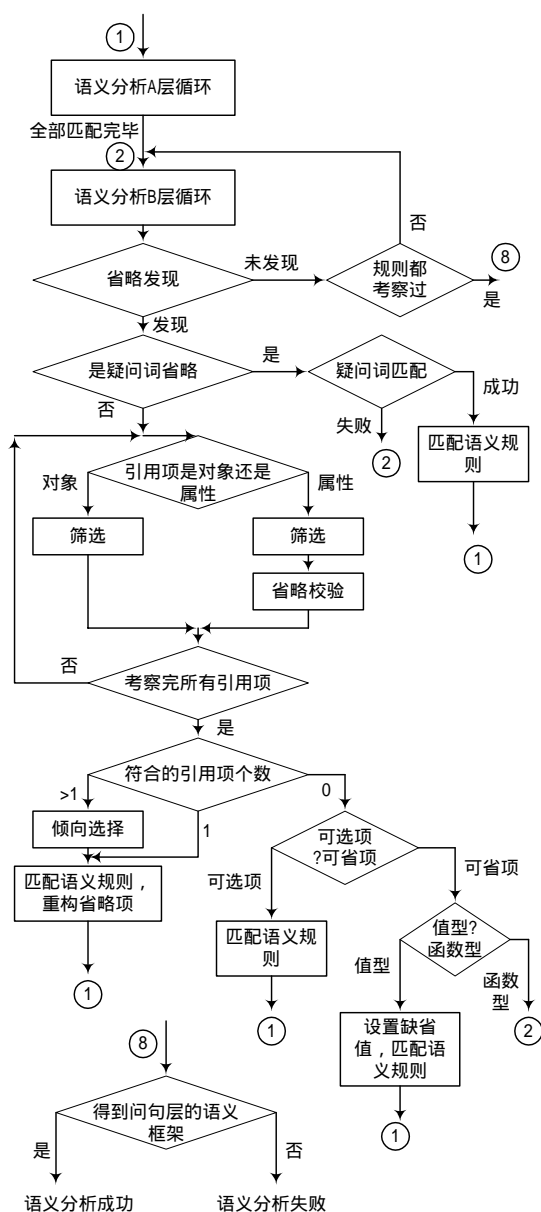


图 5-2：主题结构方法算法流程图

由于这一省略分析方法是在主题结构这一语义表示上进行的，称整个方法为主题结构方法。方法的特点是在语义知识基础上，综合会话知识和场景知识处理语境省略。其中，场景知识指关于属性优先级、可省项可选项、值型函数型省略、缺省值的知识。会话知识用于判断查询是否延续，以便省略校验时确定是否允许属性延续。

#### 5.3.4 统一的处理框架

本节从对不同省略现象进行实际分析的角度，介绍省略分析的处理框架。EasyNav 系统中的省略类型比较多，12 种主题项目中的 9 种可以被省略，涉及 64 条语义规则中的 16 条。考虑到不同省略类型可能同时出现，组合情况会很复杂。为此，处理框架采取了两个策略：1) 省略分析在语义结构自底向上的构造过程中进行，这样一次只处理一种省略；2) 不以具体省略类型为基准控制流程，而是以所省略成分的性质为基准。

##### 5.3.4.1 多约束条件查询

多约束条件查询指的是限定范围、性质、最值条件和/或地点类(或者功能)，进行的地点查询。场景中限制同一类型条件不能并列地给出多个。由于 5 种作为约束条件的项目都可以省略，涉及的规则也是最复杂的一条。

// 规则 25

场所描述 -> 属性描述(\*) + 地点类描述(.)

槽：[地点位置]=Query 地点位置([地点名])

槽：[地点名]=SelectOne([地点名集合])

槽：[地点名集合]={地点类描述}.[地点名集合] <= {属性描述}.[属性描述类型]

槽：[地点位置集合]=Query 地点位置集合([地点名集合])

关联：地点类描述 >> 地点

关联：属性描述 >> 地点

关联：[地点名] >> 地点

省略：属性描述 ~范围 >>地点 ^范围

省略：属性描述 ~性质 >>地点 ^性质

省略：属性描述 ~最值条件 >>地点 ^最值条件

省略：地点类描述 ~地点类 >>地点 ^地点类

多约束条件查询是典型的同一主题类型不同主题的例子。重列一下前述例(54)-例(55)的例子。

例(56) 用户：三十号楼附近有哪些食堂比较好？

例(57a) 用户：一号楼附近有哪些食堂？

例(56) 用户：三十号楼附近有哪些食堂比较好？

例(57b) 用户：哪些食堂比较好？

例(56) 用户：三十号楼附近有哪些食堂比较好？

例(57c) 用户：一号楼附近有哪些[食堂][比较好]呢？

例(56) 用户：三十号楼附近有哪些食堂比较好？

例(57d) 用户：[三十号楼附近]哪些食堂比较便宜？

定义相关项目的优先级如下：

Pri(范围)=5, Pri(最值条件)=4, Pri(地点类)=3

Pri(功能)=3, Pri(性质)=2

根据上一节中定义的主题变化和属性延续判定规则，我们可以对哪些约束条件得到保留，或者说哪些属性允许属性延续，作一计算。

例(57a)主题维持，新出现的属性为范围，故维持优先级为 5，性质项“比较好”不允许属性延续。

例(57b)由于约束条件集合为上一句的子集，主题转移，省略不存在。可以看到，由于约束条件集合为上一句子集，如果有省略，等于是又重问了一遍，从这个角度看，应该是问了一个新问题，而不是上一问题的延续。

例(57c)由于关键属性地点类的省略，主题延续，所有属性都允许属性延续。

例(57d)主题维持，新出现的属性为性质，故维持优先级为 2，范围项“三十号楼附近”允许属性延续。这一句与例(57a)形成一个对照。

#### 5.3.4.2 缺省值

缺省型省略指省略的信息不从语句上下文获取，而是依据场景知识获取一个缺省值，如例(58)-例(60)。

例(58) 计算机系[系馆]在哪里？

例(59) 计算机系学生[住]在哪里？

例(60a) [这里]附近有没有食堂？

然而缺省型省略往往是有歧义的，例(60)如果有上下文提到“九号楼”，则会有不同的理解：

例(60b) [九号楼]附近有没有食堂？

因此我们将缺省值也视为一种省略形式，只不过省略信息的来源不是语句上下文，而是情境上下文。缺省值是省略信息无法从语句上下文中得到时的一种替代方式。

从语句上下文中引用信息失败，一种是设缺省值，一种则取消省略，前者对应可省项，后者对应可选项。如例(61a)和例(61b)就分别对应可选项在语句上下文中引用成功和失败两种不同情况。

例(61a) [九号楼附近]哪些食堂比较便宜呢？

例(61b) 哪些食堂比较便宜呢？

对可省项进行上下文分析失败，一种情况是取缺省值，一种情况是认为分析失败。根据项目在相应语义框架中所起的作用是一个值还是一个查询函数，4.3.2 节在语义规则中区分了值型省略与函数型省略。只有对值型省略，才能在引用失败后赋予缺省值。

#### 5.3.4.3 片断

片断型省略指本身意义不完整，必须从上下文中得到省略信息才有意义的省略形式。通常表现为单独一个疑问词或是不带疑问词的短语。片断是最明显的省略，但并不等同于省略。

片断型省略是典型的句法省略，但我们在语境省略的框架下对其进行处理。原因是片断不一定仅仅是例(62)和例(63)的简单情况，还可能更复杂：可能包含若干个词，需要先进行语义分析，如例(64)；可能是新引入的属性，如例(65b)；可能导致前一句中约束条件的取消，如例(66b)。

例(62) 主楼[在哪里]呢？

例(63) [主楼]怎么走呢？

例(64) 最近的食堂[怎么走]呢？

例(65a) 用户：九号楼附近有哪些食堂？

例(65b) 用户：[九号楼附近]哪些[食堂]比较便宜？

例(66a) 用户：九号楼附近有哪些食堂比较便宜？

例(66b) 用户：[九号楼附近]哪个[食堂]最便宜呢？

这些例子表明，片断与带省略的句子难以区分，很多情况介乎两者之间。如果把片断型省略看作是一类新的省略现象，那么它要处理非片断型省略中遇到的大多数问题。因此，我们把片断型省略统一到整个语境省略的处理框架中，并且认为疑问词也可以省略，则片断中的词通过语义分析一一引用上一句中的项目，并在最后引用上一句中的疑问词，从而重构出一个完整的主题结构。

#### 5.3.4.4 零代词

零代词指的是在句中起代词作用成分的省略，如零主语、零宾语等[84]。就其性质而言，是句中存在一个主目未完全定义的谓词结构，因而需要从上下文中获取信息为主目赋值。

场景中零代词类型的省略有两种，一个是描述范围属性的规则 12，方位短语中地点的省略，一个是描述功能属性的规则 7，动词短语中施事的省略。对应省略项都是可省项，值型省略。

零代词由于省略形式比较规整，省略发现完全可以由句法规则实现。

#### 5.3.4.5 从到句

询问路线的从到句，由于句中同时出现两个地点对象而带来了复杂性。为

了在语义分析中不至混淆，源地点和目的地点都要设置句法约束条件。而在省略分析时，从上下文中寻找所省略的地点时，也要有特别的约束。

从到句中地点项省略，根据引用项是对象还是属性有不同的引用匹配条件。如果引用项为对象，匹配条件为类别相同，即类别为地点，如例(67b)。如果引用项为属性，匹配条件为与父项目关系相同，如例(68b)要求关系为源地点，例(69b)要求关系为目的地点。

从到句地点项的缺省值，倾向于“从这里到那里”的形式，如例(70b)。因此，规定目的地点不能够取缺省值，只有源地点可以取缺省值。

考虑到“从”、“到”本身也可以省略，情况更加复杂。系统中暂时规定源地点不能以对象方式引用，以便使得例(72b)能够正确分析。副作用是在例(71)有一定歧义的理解中，选取了后者例(71b')。

从到句的省略现象中，由于混杂了介词省略这样的句法省略现象，使得对例(72)的处理不理想。这表明，完整的省略处理框架，还需要把一些句法信息包括进来。

例(73)和例(74)是主题转移时允许作为对象引用的例子。例(73)由地点主题转移到路线主题，例(74)由路线主题转移到服务主题，但依然允许地点项作为对象被引用。

例(67a) 用户：九号楼在哪儿？

例(67b) 用户：从主楼[到九号楼]怎么走？

例(68a) 用户：从九号楼到图书馆怎么走？

例(68b) 用户：[从九号楼]到主楼怎么走？

例(69a) 用户：从九号楼到图书馆怎么走？

例(69b) 用户：从主楼[到图书馆]怎么走？

例(70a) 用户：九号楼在哪儿？

例(70b) 用户：[从这里][到九号楼]怎么走？

例(71a) 用户：九号楼在哪儿？

例(71b) 用户：[从九号楼]到主楼怎么走？

例(71b') 用户：[从这里]到主楼怎么走？

例(72a) 用户：九号楼在哪儿？

例(72b) 用户：[从这里][到]主楼怎么走？

例(73a) 用户：图书馆在哪里？

例(73b) 用户：[从这里][到][图书馆]怎么走？

例(74a) 用户：图书馆怎么走？

例(74b) 用户：[图书馆]附近有什么食堂？

#### 5.3.4.6 动词及其施事受事

系统中动词短语起的作用是描述地点的功能，相当于定义地点类。有三种形式：带施事、带受事、单独动词，如例(75)所示：

例(75)

计算机系办公

买文具

洗澡

为表现动词短语的不同形式，定义了若干条框架生成规则：

// 规则 7 到 10

功能描述 -> 动作(ACT) + 施事者(.)

功能描述 -> 动作(ACT, .) + 施事者

功能描述 -> 动作(ACT, .) + 受事者

功能描述 -> 动作(ACT)

规则显然存在冗余。带施事动词短语既能匹配规则 7，也能匹配规则 8。单独动词的动词短语匹配规则 10，但也会触发规则 7 进行省略发现。

这种冗余是为了在描述不同形式动词短语的同时，还能够处理动词和施事的省略而付出的代价。由于语义分析中对于多个分析结果的选择，用匹配词和引用词最多作为启发式规则，冗余并不会对分析结果的正确性造成影响。

因为受事省略在语料中出现较少，为避免带来进一步的复杂性，没有加入相应规则。

用生成规则描述动词语义结构显得描述能力不足，系统目前的语义规则不

能充分表示出动词的各个格，也不能让不同动词对应不同的格集合。

整个省略分析是以一种一致的方式实现的。不同的省略类型，归结成为具体成分的省略，或者是各种成分省略的组合，并通过自底向上、逐步重构的分析方式统一地解决。而各个省略成分又根据各自的不同性质，在流程中进入不同分支进行处理。“省略类型=>省略成分=>省略性质”的拆解过程，体现了省略分析处理框架的一般性。这使得本文提出的主题结构方法具有很强的推广性，对于场景扩充或者移植后出现的新的省略现象，很容易进行处理。

### 5.3.5 几点补充

#### 5.3.5.1 指代

由于系统中指代现象比较少，没有对指代分析专门进行处理。例(76)中的这里，规则中定义为一个直接取缺省值的成分。例(77)中的那里，可以在省略分析后得到正确的结果。

例(76) 从这里到主楼怎么走？

例(77) 从主楼到那里怎么走？

#### 5.3.5.2 现象的不同定义

EasyNav 中我们称之为省略的一系列现象，是以我们对语境省略的定义为依据所确认的。其中有些通常不归为省略现象。

如例(78)的所有格，可归为 5.1.1.1 节提到的表层指代。

例(78) 电子系的[学生宿舍]在哪里？

因为表层指代和语境省略性质上具有相似性，能够在一致的处理框架下进行计算，认为将这类现象归为省略现象更为方便。这样，在语境省略定义之下，实际上扩展了省略现象的范围。

### 5.3.5.3 文体与省略现象

省略现象作为一种上下文现象，其出现受文体影响。针对某一文体总结的省略现象和提出的分析方法，通常不适用于另一文体。如文[87]中考察的主要是叙事文体中的省略现象，与本文中考察的省略现象有很大不同。本文把这些省略现象大多归入句法省略，不进行研究。我们认为不同文体下省略现象出现不同表现是因为：1) 句子结构复杂之后，句法省略的比重将增加；2) 会话关系的主体不同，省略类型不同。

## 5.4 小结

我们把口语对话系统中出现的省略现象看作是会话现象，而非语法现象。我们对定义为语境省略的这些现象进行了研究，给出了处理办法。

在汉语口语对话系统的研究背景下，通过对省略现象的收集与分析，并仿照中心理论对指代分析与会话一致性之间关系的分析，我们综合语义、会话、场景诸因素，提出了基于主题结构的省略分析方法。在 *EasyNav* 特定的场景下，主题结构方法以一致的方式解决了省略分析中的省略发现、指代匹配、历史记录管理等问题。该方法在 *EasyNav* 中已经实现，并显示出良好的效果。

## 第六章 对话管理与查询应答

### 6.1 对话管理

对话管理模块的功能主要是根据语言理解的结果控制流程，以保证对话智能、友好地进行。设计功能有：状态切换、查询规划、上下文维护、应用查询和应答生成。后两项功能后面会单独介绍，这里简单介绍一下前三项功能。

状态切换指系统维护一个对话状态自动机，根据用户言语行为切换状态，确定系统应该回应的言语行为。该功能支持对话层次管理，使得系统具有主导能力，可以实现修补、确认等对话策略。EasyNav 场景目前主要是用户主导的连续查询，不要求系统主导，因此该功能未实现。

查询规划指系统在用户意图与任务知识冲突的情况下，重新规划查询，以求得到用户能接受的合作应答。该功能部分实现，在 6.4.2 节中会介绍。

上下文维护指系统在组织查询应答之后，更新上下文记录，以便在下一回合中由上下文分析模块使用。由于省略的局部特性，只需要保留上一句的上下文。需要记录的内容包括输入字串、语义类标记串、互连框架和主题结构。

注意存入上下文前，主题结构和互连框架都要更新，对应的主题项目要填入查询结果，对应框架中的查询函数也用查询结果代替，槽类型由函数变为值。可以认为在用户提问时，是用一个描述来指称一个对象，而在系统回答后，该对象有了确定的指云，在语义结构中也必须用确定指云来替换原先的描述，以便下文中进行引用。

### 6.2 地图知识数据库设计

地图知识数据库由与数据无关的多叉树层次结构模型表示[97]，如图 6-1 所示。

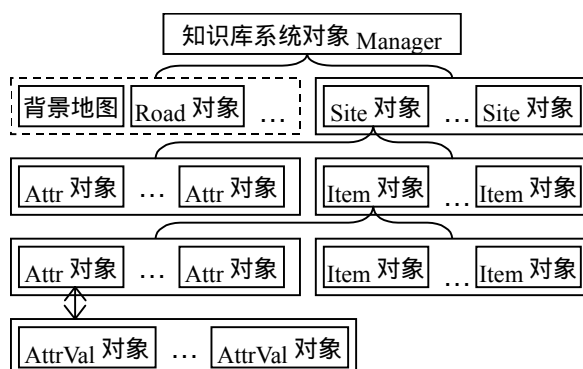


图 6-1：地图知识数据库结构

数据库以节点为单位组织，节点即为图中的 Site 对象。Site 对象分为建筑物、路口、景点及机构等四类，路口对象用于路径计算，建筑物和景点对象是进行地点查询的索引，机构对象则提供了根据机构查询地点的索引。

Site 对象包含了一组 Attr 对象和一组 Item 对象。其中 Attr 对象对应于属性，Item 对象对应于复杂性质，对应一组值。以“八号楼商店”节点为例，节点拥有 Attr 对象：便宜、好，拥有 Item 对象：卖-文具/食品/衣服。这样当查询具有相应性质的地点时，对整个 Site 对象集合进行搜索匹配，就可以得到符合性质的地点集合。

用于表示 Site 对象的数据结构有以下三个特点：

1. 它是一种递归的多叉树结构，其广度和深度均可随数据的需要而扩展。
2. 它是一种层次结构，各层次对象有不同的表达功能，子节点对象组成的链表为其父节点提供数据表示空间，父节点对象为其子节点对象提供统一的插入、删除、修改及查找等元操作。一条从根节点到叶子节点的路径（称为贯通路径）就是一条现实信息的实例。
3. 它是一种与数据无关的结构，数据的含义和相互间的关系不由数据结构确定。

模型所具有的一般性，使得知识库系统具有良好的可扩展性。

## 6.3 第一版中的查询应答

### 6.3.1 查询

第一版中语义分析得到一组预定义的句型模板中的一个所对应的语义框架，在进行查询前要转换成一个预定义的查询框架[97]。查询框架中包含了所有可能会出现的信息，是所有句型模板中所包含信息的超集，如表 6-1 所示。

表 6-1：查询框架

【QueryForm 中的域】	【注释】
QueryType	疑问词，如 where
AddOnInfo	疑问词附带信息，如 nearby
ReferenceName	参照体
FocusName	查询体
FocusCategory	查询体类别
Action	行为及动作
SubObj	行为及动作的施事/受事
AttrName	属性名
AttrVal	属性值
OriVal	方向值
MarchMode	行进方式
ItemNum	上下位项目个数
ItemArray	上下位项目链
MePos	用户所在位置

查询框架将语义框架中有关地点性质的语义结构转换成为属性值或者项目链，以对应于数据库中节点的结构，查询过程就是对节点集合进行搜索匹配。范围性质对应于节点间的关系，要先根据设定的范围限定节点集合后再进行查询。

查询框架作为语言理解与数据库查询间的接口，表示能力不够强。查询框架实际上只是罗列了所有信息，信息间有什么关系、如何使用这些信息进行查

询还要在程序中根据所对应的句型分别定义。

### 6.3.2 应答生成

从地图知识库中查询得到所需信息后，除了在地图中加亮显示相应的地点和路线外，还需要组织应答。有两种应答生成的方法，一是为每种查询模板定义一个应答模板，一是根据问句动态生成应答。系统主要采用了后一方法，只有 HOW-TO、HOW-LONG、HOW-FAR 类问句的应答是完全预先定义的模板。仿照用户的语序进行回答，可使应答显得更加自然。

动态生成应答[42]需要句法分析模块的帮助。对于汉语的特殊疑问句，由于没有语序的变化，直接把疑问词替换为查询结果，并替换第一人称和第二人称即可得到可以接受的应答。不过原句中通常存在无意义的口语附加成分，还有连词和语气词，都不应该在应答中出现。这是通过在句法树上寻找能覆盖所有语义要素的最小子句（DJ）实现的，如图 6-2 所示。

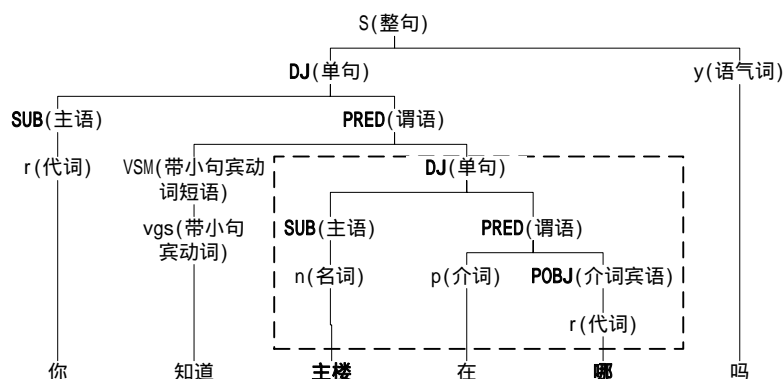


图 6-2：从句法树中提取应答模板

句法标记和关键词用黑体标出。句法树上用虚线框起的子树对应覆盖语义要素“主楼”和“哪”的最小子句。应答模板为：主楼在（answer）。

## 6.4 第二版中的查询应答

第二版中地图知识数据库没作修改，底层查询功能也未修改，但与查询模块的接口及查询的组织方式作了改变。查询接口由查询框架变为查询链，查询组织方式也由静态改变为动态。

### 6.4.1 动态查询组织

4.3.1.2 节中介绍了应答查询链，本节介绍如何使用应答查询链进行查询组织[56]。

从查询角度看，我们也把互连框架称为动态查询框架，相应称第一版中使用的查询框架为静态查询框架。动态查询框架提供了一组动态互连的框架，描述信息的能力相对于用单个框架描述句子内所有信息的静态查询框架要强得多。静态查询框架中的信息只是简单地综合在一起，动态查询框架则通过槽之间的链接、查询原语函数和前提条件，描述了信息间的关系。

动态查询框架是在语义分析过程中动态生成的，同时生成的还有框架中内含的查询链。查询链由槽之间链接、查询原语函数和前提条件等构成，查询链定义了信息的查询方式。查询链对知识库的查询提供了强有力的指引，查询模块要做的仅仅是沿着这个查询链，对知识库进行一些原子查询，以得到满足用户需求的信息。第一版中需要在程序中根据查询类型预先定义查询过程，而第二版中动态生成的查询链即定义了查询过程。这一查询组织方式我们称之为动态查询组织。

查询链作为语义分析和查询应答模块间的数据接口，很好地分隔了两个模块。语义规则设计者不需要了解数据库查询的机制细节，而只要定义将抽象的查询原语组织起来的办法。查询应答模块实现者也不需要了解如何理解用户输入得到查询表达式的过程，而只要实现对应每个查询原语的数据库查询过程。这种透明的接口也使得模块的可重用性大大提高。

系统中共定义了 32 个应答查询原语函数，其中有 3 个应答模板函数，29 个查询原语函数。查询原语作为基本的查询义表示单位，可通过组合构成复杂的查询操作。前提条件提供了将多个查询原语组合成完整查询操作的能力，槽与槽间连接则提供了先后进行多次查询操作的能力。

这种组合性也赋予了查询过程更多的灵活性。在组合查询原语进行查询操作的时候，一方面可以动态调整各查询原语的顺序，以提高查询效率，另一方面在查询失败时，可以逐条考察，找到导致查询失败的查询原语，这也就是下一节要介绍的错误处理。

### 6.4.2 应答中的错误处理

第二版中的应答，目前采用直接回答查询结果或者是固定模板应答。第一版中根据输入动态生成应答的方式，自然度高，但合作性不高，常常会在理解错误的时候给用户以正确的假象。理想的情况是能够根据语义结构生成应答，尤其是能够把从上下文中获取的信息在应答中反馈出来。由于目前系统尚不能从语义结构生成应答，因此应答生成采用最简单的方式实现。但即便是采用最简单方式实现，还是有一个问题必须解决：错误处理。

合作性作为对话系统的重要特性，要求系统在无法顺利理解用户完成操作的时候，依然能够尽量向用户提供反馈。错误处理，指的是理解或者查询失败后进行合适的应答。错误有几种情况：OOT、OOG 和约束过强。

OOT 超出任务，指用户问题超出系统任务知识的情况。如 2.3.5 节中的讨论，用一般性的语言知识补充目前任务知识的不足。如例(79)所示，语言知识中定义了“修”为功能动词 FUNC，“车”为受事 THEME，因此例(79a)能通过语义分析，但因为地图知识数据库中并没有定义修车的地点，查询不成功。例(79b)根据语言知识反馈给用户查询失败的原因，有关“修车地点”的知识超出了系统任务知识。

例(79a) 用户：自行车哪有修的？

例(79b) 系统：数据库中没有“修车地点”的信息

OOG 超出语法，指输入句型超出语义分析能力的情况。由于语义规则比较复杂不易扩充，我们加入简单的仅用于判断句型的语言知识，当检测到用户使用不支持句型的时候，提示用户换用系统所支持的句型，如例(80)。

例(80a) 用户：附近有食堂吗？

例(80b) 系统：抱歉，现在还不能够支持“有没有”这样的句型，请直接问“哪里有”

约束过强也会导致查询失败。例(81a)中对地点的查询有三重约束条件：范围为主楼附近、性质为便宜、类别为食堂。由于约束过强查询失败，则查询模块将查询操作还原成组合前的原语，并尝试放松条件，找到导致失败的约束条件，给与恰当的应答(81b)，而不是(81b')。例(82)的例子更明显，找到导致查询失败的约束，给予恰当应答(82b)，而不是(82b')。例(82a)的询问隐含了“存在好

吃的商店 ” 这一命题，回答(82b')则确认了这一隐含含义。

例(81a) 主楼附近有哪些便宜的食堂？

例(81b) 主楼附近没有食堂

例(81b') 主楼附近没有便宜的食堂

例(82a) 一号楼附近有哪些好吃的商店？

例(82b) 数据库中有关于 “ 好吃的商店 ” 的信息

例(82b') 一号楼附近没有好吃的商店

下一章中对 OOT、OOG 及错误处理还会有语料上的统计和讨论。系统指示出 OOT、OOG，以避免用户重复错误，特别是避免用户在不了解系统知识的情况下对系统不满意，这种做法对于问题宽泛型场景有重要意义。

省略和错误处理是问题的两个方面。一个始终存在的困惑是：如果只支持领域内的部分语言现象，出现领域外语言现象该怎么办？一个突出问题是：语义成分缺少该算是省略还是错误。从用户合作性角度考虑，可以认为用户并不存心造成错误，应该是共享知识不一致造成的错误，通常是用户询问了系统功能以外的问题，或是用户不了解系统常识而造成的查询错误。共享知识的不一致，一方面通过加入一般性的语言知识来使系统知识靠向用户知识；另一方面则通过错误处理使用户对系统的期望接近系统知识。

## 第七章 系统评估

### 7.1 目的与方法

系统评估是口语对话系统研究的重要环节。一方面，通过评估考察性能发现问题；另一方面，在统一语料基础上的评估提供了不同研究单位间系统性能比较的基础。

EasyNav 系统最初搭建时，并没有真实的语料，更多的是依据领域专家对系统的把握。在系统搭建出来后，需要考察领域专家对任务、语言能力的把握与真实人机交互之间有多大偏差，从而在真实语料的背景下继续完善系统。作为部分语言系统，系统中的处理策略是以常见的具有代表性的语言现象为对象的。因此，涵盖大多数语料中的语言现象，而不是涵盖语法书中大多数的语言现象，是系统评估的标准。

由于系统目前语音部分模块尚未联入，因此主要考察语言理解模块的性能，包括系统的交互成功率、语义分析成功率和上下文分析成功率等。评估的同时也对分析失败原因作了统计，以作为下一步完善的参考。

### 7.2 语料

语料来自实验室内同学和老师对系统的实际测试。测试者中不包括系统的开发人员。对测试者告知系统的功能、交互界面的使用方法，并给与几个交互的例子，要求测试者只能输入问句，尽量类似于口语对话进行连续的输入，查询内容限定在校园内地点、路线、办公、服务等方面，但查询句型不限，鼓励使用多种多样、自然的问句。

用户进行文本输入，由系统产生真实输出作为反馈。语料中删除了内容相同的重复输入。共有 13 人进行测试，共得到 514 句输入文本，平均句长 8.1 个字。扣除用户不合作的无理句子，共得到有效语料 489 句。

测试语料对系统开发人员是可见的。虽然开发人员最初的设计实现不是基于这些语料，但是测试中发现的问题对于系统完善起了很大作用：1) 发现数据

库中的数据不足或错误；2）发现不能支持或有问题的句型；3）发现有问题的上下文组合。不过也仅限于完善系统，发现数据和规则中的错误，测试语料在系统进行重设计之前并不能影响系统结构和规则体系，能够影响的主要还是应答，希望对系统不支持的句型能有友好的回答。如上所述，测试语料既非开发人员不可见的测试集，也非开发所依据的训练集，可以称之为准测试集。

EasyNav 场景没有明确的针对性，用户有很自由的选择，这使得 OOT 和 OOG 出现的可能大大增加。另外由于用户倾向于测试系统，而非使用系统，出现问题后往往会反复试验，这也导致 OOT 和 OOG 的比例增加。

语料属于人-机对话语料，且使用者由于对系统功能有一定了解，属于熟练用户。

### 7.3 评估

由于 EasyNav 为单回合交互模式，交互成功也就是在单回合交互中系统应答恰当，交互不成功则是系统应答不当。以下分成功和不成功两种情况考察系统的理解策略。

#### 7.3.1 交互成功的统计

交互成功有以下几种情况：

1. 语义分析正确，标注 S；
2. 用户输入超出任务，但可简化作任务内的查询，且得到可接受应答，标注 S-OOT；
3. 用户输入超出任务，无法查询，系统正确指出输入错误，标注 H-OOT；
4. 用户输入超出语法，无法理解，系统正确指出输入错误，标注 H-OOG。

交互成功意味着，无论用户是否得到他所想知道的内容，至少系统通过恰当的应答给与了用户明确的信息，这些信息将引导用户了解系统知识，而使得用户越来越了解系统的能力和局限，从而更好地使用系统。

下表列出了交互成功各种情况的统计：

表 7-1：交互成功统计表

	总数	交互成功	S	S-OOT	H-OOT	H-OOG
句子数	489	367	328	11	4	24
比例	----	75.05%	67.08%	2.25%	0.82%	4.91%

### 7.3.2 交互不成功的统计

交互不成功有以下几种情况：

1. 由于语义分析错误而应答错误，标注 E；
2. 由于上下文分析错误而应答错误，标注 E+；
3. 无法理解用户输入，标注 N；
4. 由于未考虑上下文而无法理解用户输入，标注 N+；
5. 用户输入超出任务，系统试图指出输入错误，但指错了，标注 EH。

E+对应的是语义成分能够正确检出，但由于上下文而产生的错误理解，通常是过度地引用上下文或者是缺省值设置有问题。如果语义成分不能够正确检出，尽管句子中有上下文现象，仍然归为 E。

N+则是设计时没有考虑到某种上下文组合情况或者是上下文维护得不好，而导致分析失败。

下表列出了交互不成功各种情况的统计：

表 7-2：交互不成功统计表

	总数	交互不成功	E	E+	N	N+	EH
句子数	489	122	46	6	53	13	4
比例	----	24.95%	9.41%	1.23%	10.84%	2.66%	0.82%

### 7.3.3 交互不成功原因统计

由于 OOT、OOG、OOV 等现象有重合，为明确造成交互不成功的主因，

我们对相应现象的标记作了限制性定义。标记 OOG 定义为出现了不支持的句型，而所查询内容在地图知识库中是有定义的。标记 OOT 定义为所查询内容在地图知识库中是无定义的，或者是支持的句型中有无定义的语义要素。标记 OOV 定义为出现了有定义语义要素的无定义形式。

造成交互不成功中 N 和 EH 的主要原因是 OOT 和 OOG。如下表的统计所示：

表 7-3：错误原因统计

	交互不成功	误输入	OOV	OOT	OOG	分析错误
句子数	122	7	10	39	55	11
比例	----	5.74%	8.20%	31.97%	45.08%	9.02%

综合交互成功句子中 OOT、OOG 的情况，得到整个语料中 OOV、OOG 出现的比例：

表 7-4：OOT、OOG 统计

	总数	OOT	OOG
句子数	489	54	79
比例	----	11.04%	16.16%

从上表可以看到：OOT、OOG 在语料中所占比例很高，合计为 27.20%。能对 OOT、OOG 进行合适的应答，对于交互成功率的提高有很大好处。目前通过对 S-OOT、H-OOT、H-OOG 的处理，使得交互成功率由单纯语义分析成功的 67.08% 提高到 75.05%，提高了 7.97%。提高比例还不高，还有很大提升余地。

交互不成功时连续出现错误是常见的现象，上一问题系统不理解，用户换种说法系统还是不理解。连续错误比例很高，在所有错误中占 50%。同时连续错误也扩大了系统存在的问题，使得用户对系统不满意程度增加。我们在 6.4.2 节考虑错误处理，就是试图在发生错误后尽量引导用户了解系统的局限，避免连续出错。

交互不成功原因中没有数据库提供信息错误，这是因为我们把测试语料指

示出的数据库错误已经作了更正，统计结果是更正后进行的。扣除数据库错误的原因，我们把系统评估的重点放在语言理解能力上。

#### 7.3.4 语义分析成功率

表 7-1 中语义分析成功率 67.08%，如果扣除误输入、OOV、OOT、OOG 等情况，则语义分析成功率上升为 96.76%。OOV 和 OOG 是由于语言知识的不足引起的，考虑 OOV 和 OOG 之后，语义分析成功率为 76.64%。

在语料中，有些句子是有歧义的，目前都只按某一含义处理，并标注为语义分析正确。如例(83b)是陈述句，可理解为询问到西门的路线，系统按照(83b')去理解，作为片断型省略，理解为询问路线的长度。尽管意图理解不确切，如果应答得好，也可以让用户满意。

例(83a) 九号楼有多远？

例(83b) 我想去西门

例(83b') 西门呢？

语料中还有语义分析正确但应答不是很好的例子。如对例(84a)的回答，并没有列举所有的食堂和餐厅，而是随机选一作为应答。

例(84a) 用户：你知道哪里可以吃饭？

例(84b) 系统：清东餐厅，您的北边、六号楼的前边

#### 7.3.5 上下文分析成功率

我们首先对语料中上下文现象出现的次数进行统计。虽然缺省值的处理被纳入省略分析框架之中，但为了便于分析，我们将其分开来统计。则有表 7-5：

表 7-5：上下文处理统计

	总数	语句上下文	分析正确	缺省值	分析正确
句子数	489	104	61	71	63
比例	----	21.27%	12.47%	14.52%	12.88%

表中“语句上下文”指的是省略指代分析从语句上下文中引用对象的情况，则上下文现象出现的频度为 21.27%。如果连同从情境上下文中获取信息的句子进行统计，上下文现象出现频度为 35.79%。上下文现象的大量出现表明了对其分析处理的必要性。

下面我们对上下文分析错误原因作一统计。由于缺省值是在省略分析框架中进行的处理的，因此合并起来统一分析。

表 7-6：上下文分析统计

	上下文	错误总数	本句不合法	上下文无效	处理错误
句子数	175	51	40	5	6
比例	----	29.14%	22.86%	2.86%	3.43%

上下文分析错误率高达 29.14%，但很大程度上是因为本句出现误输入、OOV、OOT 或者 OOG 造成的，扣除本句不合法的情况后，错误率降到 8.15%，也即成功率为 91.85%。错误中将近一半是因为上下文维护的问题，通常是上一句没有理解而无法保留相应语义结构的缘故，另一半则是省略分析算法中出现的其它问题。

另外要注意的是虽然语料中没有出现倾向选择的歧义，但省略校验、缺省值使用都出现一些带歧义的句子，甚至单从句子本身无法判断该选择哪一种含义。对这些句子，都算在上下文分析正确的统计中。目前选择其一作为正确含义，通过应答给用户以反馈。

## 第八章 结论

近十年来,随着语音识别技术的成熟,语音理解逐渐成为语音处理领域的一个重要研究方向。让人-机对话完全达到人-人对话那样自然,固然不可能,但在限定任务(可以是内部具有复杂关系、多主题的任务)背景下,在双方都具有合作意愿的前提下,令人-机对话达到相当的自然度,却是可能达到的目标。口语对话系统的研究,从早期的菜单式电话系统,到单主题表单式系统,再到多主题知识主导系统,尽管是在场景受限的约束之下,但始终向着令对话过程更自然、用户具有更多主导能力这个方向发展。我们的目的就是为复杂任务背景提供自然对话的理解技术。具体来说,对于领域受限的系统,考察需要哪些知识、如何利用这些知识,才能提高理解和对话能力,使对话系统真正能够实用。

在此目标驱动之下,本文重点研究语义分析与表示和省略分析这两个课题。

(一). 提出互连框架、查询链和主题结构进行语义表示,提出互连框架方法进行语义分析。互连框架方法的特点是:1)互连框架等语义表示依据意义组合原则提出,具有比较强的描述能力;2)语义表示单位具备意义计算能力,分析过程中能够动态生成整句意义计算信息;3)双层多遍的语义分析流程与上下文分析紧密耦合,框架互连的形式使得上下文分析所需要的部分语义表示易于抽取。

互连框架方法能分析复杂的语义结构,能分析包含省略现象的连续提问。互连框架内含的应答查询链提供对整句意图的计算,指导后续的查询与应答处理。互连框架方法在语义表示能力上优于其它基于语义框架、格框架的语义分析表示方法,表现在:1)其它方法中,框架不包含计算信息,从而使后续的查询与应答处理只能使用预定义格式信息而无法利用语义结构信息,削弱了语义表示的描述能力;2)其它方法并未提供语义分析和上下文分析的紧密耦合,也未提供上下文分析所需的部分语义表示。

(二). 提出语境省略的概念。针对汉语省略现象特点提出,强调:1)基于语义知识进行省略检测;2)立足于当前句进行省略重构。并在比较指代和省略现象异同的基础上,提出省略发现、筛选、省略校验和倾向选择四步骤框图以概括语境省略分析的处理要素。到目前为止,国内外汉语省略方面的研究很

少，汉语口语对话下的省略研究更少，对汉语口语对话下省略计算方法进行研究的几乎没有，主题结构方法在这方面的研究具有原创性。

(三). 提出主题结构方法进行省略分析。特点是：1) 符合语境省略概念，适合处理汉语口语对话中出现的省略现象；2) 对主题变化及其对约束条件延续的作用进行建模；3) 通过“省略类型=>省略成分=>省略性质”的拆解，建立一般性的处理框架。主题结构方法能分析 EasyNav 场景内各种省略类型及其组合。处理框架具有一般性，容易进行扩展以支持新的省略类型。主题变化建模考察多主题、主题变化频繁场景中，主题变化与省略分析的联系，这一问题超出了大多数表单式单主题系统的研究范围，具有前瞻性。

本文其它的工作围绕设计实现口语对话系统 EasyNav 而展开。包括：对系统流程的整体设计、基于 GLR 的句法分析、对话管理、应用查询、应答生成。系统在文本输入输出条件下能工作，从而使前述研究工作不只停留在理论层面，而是实际实现，并在实践过程中进一步得到验证和完善。

EasyNav 使用校园导航场景，是真实场景，使用者由于对校园信息比较熟悉也相当于真实用户，在这种真实条件下进行的测试得到的系统性能评估比较可信。评估表明：扣除场景外输入，语义分析成功率为 96.76%，上下文分析成功率为 91.85%，说明本文提出的互连框架方法和主题结构方法是有效的；场景外输入在语料中占 27.20%，说明语法覆盖率问题还是自然对话场景的一大问题。

## 参考文献

- [1] Zue V W, Glass J R. Conversational interfaces: advances and challenges. Proceedings of the IEEE, 2000, 88(8): 1166~1180
- [2] Stephanie S. TINA: A natural language system for spoken language applications. Computational Linguistics, 1992, 18(1): 61~86
- [3] Bennacef S, Bonneau-Maynard H, Gauvain J L, et al. A spoken language system for information retrieval. In: Proc. of ICSLP. 1994. 1271~1274
- [4] Strom, N, Seneff S. Intelligent barge-in in conversational systems. In: Proc. of ICSLP. 2000. (2): 652~655
- [5] Potamianos A, Ammicht E, Kuo H. Dialogue management in the BELL Labs Communicator system. In: Proc. of ICSLP. 2000. (2): 603~606
- [6] Larsson S, Santamarta L, Jonsson A. Using the process of distilling dialogues to understand dialogue systems. In: Proc. of ICSLP. 2000. (3): 374~377
- [7] Fafiotte G, Zhai J. From multilingual multimodal spoken language acquisition towards on line assistance to intermittent human interpreting: sim\*, a versatile environment for SLP. In: Proc. of ICSLP. 2000. (2): 110~113
- [8] Walker M A, Litman D J, Kamm C A, et al. Evaluating spoken dialogue agents with PARADISE: Two case studies. Computer Speech and Language, 1998, 12(4): 317~347
- [9] Cole R, Villiers J, Shobaki K, et al. Demonstration of dialogue tools in the CSLU Toolkit. In: Proc. ESCA Workshop on Interactive Dialogue in Multi-Modal Systems. Kloster Irsee, Germany, 1999.
- [10] Glass J, Weinstein E. SpeechBuilder: Facilitating spoken dialogue systems development. In: Proc. of EUROSPEECH. 2001. 1335~1338
- [11] Nouza T, Nouza J. Graphic platform for designing and developing practical voice interaction systems. In: Proc. of EUROSPEECH. 2001. 1287~1290
- [12] Mariani J, Lamel L. An overview of EU programs related to conversational/ interactive

- systems. In: Proceedings of DARPA Broadcast News Transcription and Understanding Workshop. 1998. 247~253
- [13] Peckham J. A new generation of spoken dialogue systems: results and lessons from the SUNDIAL project. In: Proc. of EUROSPEECH. 1993. 33~40
- [14] Gauvain J L, Bennacef S, Devillers L, et al. Spoken language component of the MASK kiosk. In: Varghese K, Pfleger S, eds. Human Comfort & Security of Information Systems, Springer, 1997. 93~103
- [15] Lamel L F, Bennacef S K, Rosset S, et al. The LIMSI RailTel system: Field trial of a telephone service for rail travel information. *Speech Communication*, 1997, 23(1-2): 67~82
- [16] den Os E, Boves L, Lamel L, et al. Overview of the ARISE project. In: Proc. of EUROSPEECH. 1999. 1527~1530
- [17] Lamel L, Rosset S, Gauvain J L, et al. The LIMSI ARISE system for train travel information. In: Proc. of ICASSP. 1999. 501~504
- [18] Sturm J, den Os E, Boves L. Issues in spoken dialogue systems: experiences with the Dutch ARISE system. In: Proc. ESCA Workshop on Interactive Dialogue in Multi-Modal Systems. Kloster Irsee, Germany, 1999. 1~4
- [19] Pallett D, Fiscus J, Fisher W, et al. Benchmark tests for the ARPA spoken language program. In: Proc. ARPA Spoken Language Systems Technology Workshop. 1995. 5~36
- [20] DARPA Communicator Website: <http://fofoca.mitre.org/>
- [21] Seneff S, Lau R, Polifroni J. Organization, communication, and control in the GALAXY-II conversational system. In: Proc. of EUROSPEECH. 1999. 1271~1274
- [22] Walker M, Aberdeen J, Boland J, et al. DARPA Communicator dialog travel planning systems: the June 2000 data collection. In: Proc. of EUROSPEECH. 2001. 1371~1374
- [23] Glass J, Flammia G, Goodline D, et al. Multilingual spoken language understanding in the MIT VOYAGER system. *Speech Communication*, 1995, 17(1-2): 1~18
- [24] Zue V, Seneff S, Glass J, et al. JUPITER: A telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, 2000, 8(1): 100~112

- [25] Zue V, Seneff S, Polifroni J, et al. PEGASUS: A spoken dialogue interface for on-line air travel planning. *Speech Communication*, 1994, 15(3-4): 331~340
- [26] Seneff S, Polifroni J. Dialogue management in the Mercury flight reservation system. In: *Proc. of ANLP-NAACL 2000 Workshop on Conversational Systems*. Seattle, WA, 2000. 11~16
- [27] Seneff S, Chuu C, Cyphers D S. Orion: From on-line interaction to off-line delegation. In: *Proc. of ICSLP*. 2000. (2): 142~145
- [28] Meng H, Busayapongchai S, Glass J, et al. Wheels: A conversational system in the automobile classifieds domain. In: *Proc. of ICSLP*. 1996. 542~545
- [29] Seneff S, Polifroni J. A new restaurant guide conversational system: Issues in rapid prototyping for specialized domains. In: *Proc. of ICSLP*. 1996. 665~668
- [30] Lau R, Flammia G, Pao C, et al. WebGalaxy – Integrating spoken language and hypertext navigation. In: *Proc. of EUROSPEECH*. 1997. 883~886
- [31] Allen J F, Schubert L K, Ferguson G, et.al. The TRAINS project: A case study in building a conversational planning agent. The University of Rochester, TRAINS-TN-94-3, 1994
- [32] Ferguson G, Allen J. TRIPS: An intelligent integrated problem-solving assistant. In: *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*. Madison, WI, 1998. 567~573
- [33] Carlson R, Hunnicutt S, Gustafson J. Dialogue management in the Waxholm system. In: *Proc. ESCA/ETRW on Spoken Dialogue Systems*. Vigso, 1995. 137~140
- [34] Gustafson J, Elmberg P, Carlson R, et al. An educational dialogue system with a user controllable dialogue manager. In: *Proc. of ICSLP*. 1998. 33~37
- [35] Gustafson J, Lindberg N, Lundeberg M. The August spoken dialogue system. In: *Proc. of EUROSPEECH*. 1999. 1151~1154
- [36] Gustafson J, Bell L, Beskow J, et al. AdApt – a multimodal conversational dialogue system in an apartment domain. In: *Proc. of ICSLP*. 2000. (2): 134~137
- [37] Ahrenberg L, Dahlback N, Jonsson A. Discourse representation and discourse management for a natural language dialogue system. In: *Proceedings of the Second Nordic Conference on Text Comprehension in Man and Machine*. 1990.

- [38] Dahlback N, Flycht-Eriksson A, Jonsson A, et al. An architecture for multi-modal natural dialogue systems. In: Proc. ESCA Workshop on Interactive Dialogue in Multi-Modal Systems. Kloster Irsee, Germany, 1999.
- [39] Chu-Carroll J, Carpenter R L. Vector-based natural language call routing. Computational Linguistics, 1999, 25(3): 361~388
- [40] Gorin A L, Riccardi G, Wright J H. How may I Help You? Speech Communication, 1997, 23(1-2):113~127
- [41] Huang C, Xu P, Zhang X, et al. LODESTAR: A mandarin spoken dialogue system for travel information retrieval. In: Proc. of EUROSPEECH. 1999. 1159~1162
- [42] 黄寅飞, 郑方, 燕鹏举, 等. 校园导航系统 EasyNav 的设计与实现. 中文信息学报, 2001, 15(4): 35~40
- [43] 燕鹏举, 陆正中, 邬晓钧, 等. 航班信息系统 EasyFlight. 见: 第六届全国人机语音通讯学术会议 (NCMMSC'6). 2001. 115~118
- [44] Wang X F, Du L M. The design of dialogue management in a mixed initiative Chinese spoken dialogue system engine. In: Proc. of ISCSLP. 2000. 53~56
- [45] 何伟, 袁保宗, 林碧琴, et al. 面向导游任务的人机口语对话系统的研究与实现. 见: 第六届全国人机语音通讯学术会议 (NCMMSC'6). 2001. 97~101
- [46] Varile G B, Zampolli A (managing editors). Survey of the state of the art in human language technology. Cambridge [England]; New York : Cambridge University Press, 1997.
- [47] Glass J. Challenges for spoken dialogue systems. In: Proc. IEEE ASRU Workshop. Keystone, CO, 1999.
- [48] Lee C H. Spoken dialogue processing towards telecommunication applications. In: Proceedings of International Symposium on Spoken Dialogue (ISSD). 1998. 35~50
- [49] Jurafsky D, Martin J H. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. Upper Saddle River, N.J: Prentice Hall, 2000.
- [50] Allen J. Natural language understanding (2nd ed.). Redwood City, Calif. : Benjamin/Cummings Pub. Co. 1995.

- [51] Huang Y, Zheng F, Xu M, et al. Language understanding component in Chinese dialogue system. In: Proc. of ICSLP. 2000. (3): 1053~1056
- [52] Su Y, Zheng F, Huang Y. Design of a semantic parser with support to ellipsis resolution in a Chinese spoken language dialogue system. In: Proc. of EUROSPEECH. 2001. 2161~2164
- [53] 黄寅飞, 郑方, 苏毅, 等. 口语对话系统 EasyNav 中的语义表示. 见: 第六届全国人机语音通讯学术会议 (NCMMSC'6) . 2001. 103~106
- [54] 黄寅飞, 郑方, 苏毅, 等. 口语对话系统中 EasyNav 的省略分析. 清华学报, 2002, 42(1): 88~91
- [55] Huang Y, Zheng F, Su Y, et al. A Theme Structure method for the ellipsis resolution. In: Proc. of EUROSPEECH. 2001. 2153~2156
- [56] Li F, Zheng F, Wu W, et al. Dynamic query organization and response generation in spoken dialogue system. In: Proc. of ICCPOL. 2001.
- [57] 李芳. EasyNav2.0 中的动态查询组织和应答生成: [硕士学位论文]. 北京: 清华大学计算机系, 2001
- [58] 邹崇理. 逻辑、语言和蒙太格语法. 北京: 社会科学文献出版社, 1995.
- [59] Nakagawa S. Architecture and evaluation for spoken dialogue systems. In: Proceedings of International Symposium on Spoken Dialogue (ISSD). 1998. 1~8
- [60] 方立. 逻辑语义学. 北京: 北京语言文化大学出版社, 2000.
- [61] 周强, 俞士汶. 汉语短语标注标记集的确定. 中文信息学报, 1996, 11(4): 1~11
- [62] Tomita M. Generalized LR Parsing. Kluwer Academic Publishers, 1991. 1~15
- [63] 陈火旺, 钱家骅, 孙永强. 程序设计语言编译原理 (第二版). 北京: 国防工业出版社, 1984.
- [64] Seneff S, Polifroni J. Formal and natural language generation in the Mercury conversational system. In: Proc. of ICSLP. 2000. (2): 767~770
- [65] Baptist L, Seneff S. GENESIS-II: a versatile system for language generation in conversational system applications. In: Proc. of ICSLP. 2000. (3): 271~274
- [66] 徐烈炯. 语义学 (第二版). 北京: 语文出版社, 1995.

- [67] 曹逢甫. 主题在汉语中的功能研究. 北京: 语文出版社, 1995.
- [68] Johnson F C. A classification of ellipsis based on a corpus of information seeking dialogues. *Information Processing & Management*, 1994, 30(3): 315~325
- [69] Greene J, Coulson M. *Language understanding : current issues* (2nd ed.). Buckingham [England] ; Philadelphia : Open University Press, 1995.
- [70] Carbonell J G, Brown R D. Anaphora resolution: A multi-strategy approach. In: COLING'1988. 96~101
- [71] Kamp H, Reyle U. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*. Dordrecht ; Boston : Kluwer Academic, 1993.
- [72] Grosz B J, Joshi A K, Weinstein S. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 1995, 21(2): 203~225
- [73] Allen J. *Natural language understanding*. Menlo Park, Calif. : Benjamin/ Cummings Pub. Co. 1987.
- [74] Hobbs J R. Resolving pronoun references. *Lingua*, 1978, 44: 311~338
- [75] Lappin S, Leass H J. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 1994, 20(4): 535~561
- [76] Strube M. Never look back: an alternative to centering. In: ACL'36&ICCL'17. 1998. (2): 1251~1257
- [77] Mitkov R, Boguraev B, Lappin S. Introduction to the special issue on computational anaphora resolution. *Computational Linguistics*, 2001, 27(4): 473~477
- [78] Churcher G, Wyard P. Developing robust, user-centred multimodal spoken language systems: the MUeSLI Project. In: *Proc. of ICSLP. 2000.* (3): 1045~1048
- [79] Johnson K. What VP ellipsis can do, what it can't, but not why. In: Baltin M, Collins C, eds. *The Handbook of Contemporary Syntactic Theory*. 2001. 439~479
- [80] Lappin S, Gregory H. A computational model of ellipsis resolution. In: *Proc. Conf. on Formal Grammar, ESSLLI'1997.*
- [81] Lappin S, Shih H H. A generalized reconstruction algorithm for ellipsis resolution. In:

- COLING'1996.
- [82] Merchant J. The syntax of silence. Sluicing, islands, and identity in ellipsis: [PhD. Diss.]. University of California, Santa Cruz, 1999
- [83] Kennedy C, Merchant J. Attributive comparative and bound ellipsis. Linguistic Research Center Report LRC-97-03, University of California, Santa Cruz, 1997
- [84] Walker M, Iida M, Cote S. Japanese discourse and the process of centering. Computational Linguistics, 1994, 20(2): 193~232
- [85] Palomar M, Ferrandez A, Moreno L, et al. An algorithm for anaphora resolution in Spanish texts. Computational Linguistics, 2001, 27(4): 545~567
- [86] Ginzburg J, Gregory H, Lappin S. SHARDS: Fragment resolution in dialogue. In: Proceedings of the Fourth International Conference on Computational Semantics. Tilburg. 156~172
- [87] 黄南松. 现代汉语叙事体语篇中的成分省略. 中国人民大学学报, 1996, (5): 75~80
- [88] Dalrymple M, Shieber S, Pereira F. Ellipsis and higher-order unification. Linguistics and Philosophy, 1991, 14(4): 399~452
- [89] 中国社会科学院语言研究所“汉语运用的语用原则”课题组. 语用研究论集. 北京: 北京语言学院出版社, 1994.
- [90] Finch G. Linguistics terms and concepts. New York: St. Martin's Press, 2000.
- [91] Kehler A. Coherence, reference, and the theory of grammar. In: The 7th Annual CSLI Workshop on Logic, Language, and Computation. 1998.
- [92] Grosz B J, Sidner C L. Attention, intentions, and the structure of discourse. Computational Linguistics, 1986, 12(3): 175~204
- [93] Hobbs J R. Coherence and coreference. Cognitive Science, 1979, 3: 67~90.
- [94] Hobbs J R, Stickel M E, Appelt D E, et al. Interpretation as abduction. Artificial Intelligence, 1993, 63: 69-142.
- [95] 吕叔湘. 通过对比研究语法. 见: 马庆株编. 语法研究入门. 北京: 商务印书馆, 1999. 13~27
- [96] Greenbaum S. The Oxford English grammar. New York: Oxford University Press, 1996.

- [97] Yan P, Zheng F, Xu M, et al. Word-class stochastic model in a spoken language dialogue system. In: Proc. of ISCSLP. 2000. 141~144

---

## 致 谢

感谢我的导师吴文虎教授。在我研究和论文工作中，吴老师耐心地启发我提炼观点，帮助我修改润色文字。吴老师积极认真的态度、谦和仁厚的风范对我有深远的影响。

感谢我的副导师郑方老师。郑老师对我的选题和研究工作给予了大量指导，他对于国际学术动态的把握令我受益非浅。郑老师勤奋务实的作风、团队组织和项目管理的能力都值得我学习。

感谢徐明星老师、方棣棠教授和李树青教授的建议和指导。

特别感谢我的父母和家人的支持与鼓励。

感谢 EasyNav 系统第一版的主要合作者：燕鹏举，以及其他合作者：徐明星老师、张继勇、武健同学。感谢 EasyNav 系统第二版的主要合作者：苏毅、李芳，以及其他合作者：张华、李山山。

感谢何磊经常倾听我的想法。感谢中科院自动化所的徐为群对我学术上的交流与帮助。感谢实验室内各位同学：宋战江、王帆、杨大利、张国亮、李净、罗春华、邬晓钧、吴根清、熊振宇、孙辉、陆正中、金陵、余斌、张欣研。

---

## 个人简历

黄寅飞生于 1974 年 2 月 15 日,于 1997 年 7 月获得清华大学计算机科学与技术系学士学位,同年 9 月推荐为清华大学计算机科学与技术系直读博士生。

## 研究成果

- 1999.6-2001.4, 作为 EasyNav 组长,组织完成自选项目“校园导航口语对话系统 EasyNav”
- 1998.8-1999.1, 开发桌面命令导航系统 EasyCmd

## 学术论文

- 黄寅飞,吴文虎. 汉语语音识别技术综述. 中国计算机用户,2000,(11)
- Huang Y, Zheng F, Wu W. EasyCmd: Navigation by voice commands. In: Proc. of ISCSLP. 2000. 145~148
- Yan P, Zheng F, Xu M, et al. Word-class stochastic model in a spoken language dialogue system. In: Proc. of ISCSLP. 2000. 141~144
- Huang Y, Zheng F, Xu M, et al. Language understanding component in Chinese dialogue system. In: Proc. of ICSLP. 2000. (3): 1053~1056
- 黄寅飞,郑方,燕鹏举,等. 校园导航系统 EasyNav 的设计与实现. 中文信息学报,2001,15(4): 35~40
- Huang Y, Zheng F, Su Y, et al. A Theme Structure method for the ellipsis resolution. In: Proc. of EUROSPEECH. 2001. 2153~2156
- Su Y, Zheng F, Huang Y. Design of a semantic parser with support to ellipsis resolution in a Chinese spoken language dialogue system. In: Proc. of EUROSPEECH. 2001. 2161~2164
- Li F, Zheng F, Wu W, et al. Dynamic query organization and response generation in spoken dialogue system. In: Proc. of ICCPOL. 2001.
- 黄寅飞,郑方,苏毅,等. 口语对话系统 EasyNav 中的语义表示. 见:

---

第六届全国人机语音通讯学术会议 (NCMMSC'6) . 2001. 103~106

- 黄寅飞, 郑方, 苏毅, 等. 口语对话系统中 EasyNav 的省略分析. 清华学报, 2002, 42(1): 88~91