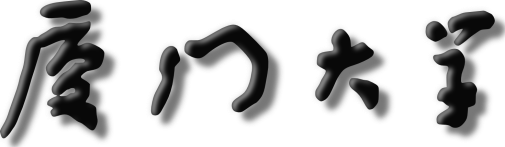
**学校编码： 10384 分类号 T192 密级 公开 学号：23020081152XXX UDC 168**



**博 士 学 位 论 文**

**动机驱动的认知对话系统的关键技术研究**

**Motivation-driven Cognitive Methodology for Natural Language Dialogue**

练 睿 婷

**指导教师姓名： 周昌乐 教授**

**专 业 名 称： 人工智能基础**

**论文提交日期： 2016 年 6 月**

**论文答辩时间： 2016 年 7 月**

**学位授予日期： 年 月**

**答辩委员会主席：**

**评 阅 人：**

**2016 年 6 月**

厦门大学学位论文原创性声明

本人呈交的学位论文是本人在导师指导下，独立完成的研究成果。 本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中 以适当方式明确标明，并符合法律规范和《厦门大学研究生学术活动规 范（试行）》。

另外，该学位论文为（ ）课题（组）的研究成果，获得（ ）课题（组）经费或实验室的资助，在（ ）实验室完成。（请在以上括号内填写课题或课题组负责人或实验室名称，未有此项声明内容的，可以不作特别声明。）

声明人（签名）：

年 月 日

厦门大学学位论文著作权使用声明

本人同意厦门大学根据《中华人民共和国学位条例暂行实施办法》 等规定保留和使用此学位论文，并向主管部门或其指定机构送交学位论 文（包括纸质版和电子版），允许学位论文进入厦门大学图书馆及其数 据库被查阅、借阅。本人同意厦门大学将学位论文加入全国博士、硕士 学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于：

（ 　）1、经厦门大学保密委员会审查核定的保密学位论文，于 年 月 日解密，解密后适用上述授权。

（ ）2、不保密，适用上述授权。

（请在以上相应括号内打“√” 或填上相应内容。保密学位论文应 是已经厦门大学保密委员会审定过的学位论文，未经厦门大学保密委员 会审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开 学位论文，均适用上述授权。）

声明人（签名）：

年 月 日

# 

# 摘 要

使机器拥有自然语言交流能力是人工智能的终极目标之一，自图灵测试被提出以来，机器的对话能力也一直被当作衡量系统智能程度的标准。近年来，对话系统更是成为人工智能在商业领域的热点，各种对话系统纷纷发布并，如苹果的Siri，微软的Cortana和小冰，小i和图灵等等。尽管数十年来，NLP与AI研究在语音识别、语音合成、机器翻译、信息检索、浅层语义、情感分析等领域取得巨大进展，但对应用于对话系统的认知对话管理、语义语用等深层理解领域仍没有根本性的突破。即使目前表现最好的对话系统，也仍然依赖于表层的语言分析和简单的启发性知识，而不考虑语义和语用等信息，因此其表现仍然与人类的对话表现有巨大差异。

构建一个人类水准的对话系统是一项艰巨而长远的任务，其深度和内容都已经远超本文所能涵盖的范围。本文针对当前对话系统中的认知技术研究的不足，深入研究了认知对话系统中所需涉及的认知技术的范畴，在这些认知需求的基础上提出了一个动机驱动的认知对话控制模型，我们认为，一个自然的人机对话系统，必须遵循以下四个原则：知识深层表示、不确定性逻辑推理、语言与涉身知识的融合以及动机驱动的对话控制。

在知识深层表示方面，针对传统对话系统中知识表示的不足所引起的表达能力和计算效率问题，本文使用了一个基于超图的知识表示体系，该体系既能储存丰富复杂的语义信息且能灵活操作。在该知识表示体系的基础上，本文设计和实现了能将自然语言转换成基于该表示的语义逻辑形式的自然语言理解框架，使得该对话系统能够从自然语言中获取浅层的句法信息之外的深层语义信息，并用于对话认知过程中；此外，本文还设计并实现了一个基于超图匹配的将抽象的语义超图表示转换成自然语言的表层生成框架。

在不确定性逻辑推理方面，鉴于能进行自然的人机对话的系统必须具备对不确定性知识的处理能力和在非精确条件下推理和决策控制的能力，本文在概率逻辑网络的逻辑架构上应用不确定性逻辑推理，实现了在自然语言语句上执行概率性常识推理，使得人机对话过程中系统能借助常识来理解模糊或不确定的输入。

在语言与涉身知识的融合方面，鉴于语言的交互不仅仅是语言层面上的交互，还需要理解语言背后涉及到的涉身交互，本文以言语行为理论作为指导，将对话中的每一句话语看成是成一个“言语行为”，即包含一些独特的语言属性，还包含一些涉及到言语行为和其他类型行为的语用属性，即“言外之意”，从而将语言和涉身知识融合并将言语所对应的行为类型作为对话策略的选择依据之一。

在动机驱动的对话控制方面，鉴于目前的对话系统中大多使用基于简单的浅层语言信息提示来回应用户输入，本文认为认知对话系统应该具备一个动机系统，使得对话过程能够按照系统本身的动机出发来完成认知型的智能对话，而不仅仅作为“执行主体”来完成用户指派的任务，因此本文引入了心理学上知名的动机驱动的情感评估模型Psi，并改进该模型使其更适合用于认知对话系统中，通过动机驱动的模型来组织和管理对话，使得对话过程中系统能根据其当前的状态和动机来选择最合适的对话行为作为应答。

本文着眼于未来对话系统中认知技术的发展，提出了一个全新的动机驱动的认知对话系统模型，并以言语行为理论作为指导将语言和涉身知识融合起来使得对话过程更拟人更自然。本文对对话系统的认知技术进行了深入研究和有效的探索，为实现更逼真更自然的对话系统奠定了一定的基础。

**关键词：**言语行为；语言推理；对话系统

# 

# Abstract

Emulation of the human capability for natural language dialogue has been central to the enterprise of Artiﬁcial Intelligence since the very beginning of the ﬁeld, going back to Alan Turing’s identiﬁcation of fully human-like dialogue as a litmus test for successful AI (i.e. what is now called the “Turing Test”). In recent years, natural language dialogue has also become important in the commercial AI sphere, with high proﬁle oﬀerings such as Apple’s Siri, and Microsoft’s Cortana and Xiaoice. There have been dramatic recen- t advances in many related, supporting areas of computational linguistics and AI, such as speech recognition and generation, machine translation, information retrieval, parsing, semantic analysis, sentiment and opinion mining, and many others. In spite of this relat- ed progress, however, there have been no truly fundamental breakthroughs in the area of dialogue systems proper. Even the best existing systems operate mainly based on surface- level linguistic cues and simple heuristics, rather than based on richly mapping and re- sponding to the semantic and pragmatic intentions implicit in the human portions of the dialogue. The result of these internal deﬁcits in semantics and pragmatics are externally manifested as dialogue performance that generally becomes tiresome to the human user after a brief period.

Creating truly human-level dialogue systems is a huge project, as witnessed by the large teams and large sums of money that technology ﬁrms have recently devoted to creating dialogue systems with rather severely limited functionality. Our goal in this thesis is not to fully solve the problem, but to lay a new conceptual and practical foundation for ongoing work. The key hypothesis underlying our work is that, to achieve a broadly useful level of functionality (and as well to ultimately achieve a truly human-level functionality), a dialogue system must be constructed according to the following four principles：

* **deep representation** ： dialogue should be fundamentally founded on mapping natu-ral language utterances into a rich, and ﬂexibly manipulable, semantic representation–in which an utterance is given a ”deep representation” that is largely separate from the properties of the surface-level language involved in an utterance
* **uncertain logical inference：** dialogue should involve inference that is both logical (involving inferential steps such as deduction, induction and abduction) and uncer- tain (encompassing e.g. probabilistic and fuzzy aspects of uncertainty).
* **foundation in speech act theory：** linguistic interactions should be understood as embedded in a larger matrix of embodied interactions –, so that each utterance is understood as a “speech act” with some uniquely linguistic properties, and some pragmatic properties that are shared by speech acts and other sorts of actions.
* **motivated dialogue control：** dialogue control (the choice of what to say when) should be founded on a model of intelligent agent motivation： the system should choose what to say based on its underlying motivations, rather than based on simple cue-response patterns.

With the above four principles in mind, our research has been primarily explored them and have achieved to design and implement and a “cognitive dialogue model” embodying the four principles serving as the foundation of our investigation. The major works and contributions of this thesis can be summarized as follows：

* given a novel formulation of dialogue control in terms of Speech Act Theory and the Psi model of action selection.
* demonstrated the utilization of uncertain logical reasoning (via the Probabilistic Logic Networks logic framework) to perform commonsense logical inference from statements uttered in natural language.
* created speciﬁc mechanisms for question-answering based on deep representations of natural language knowledge and queries, and demonstrated the utilization of these mechanisms for queries on a large common sense knowledge base (Simple English Wikipedia).
* designed and implemented a novel framework using hypergraph pattern matching to translate the output of a dependency parser into an abstract logic formalism.
* designed and implemented a novel algorithm for surface realization, utilizing hy- pergraph pattern matching to generate natural language sentences corresponding to semantic hypergraphs, via leveraging knowledge obtained from sentences the sys- tem has previously interpreted

Via these speciﬁc accomplishments, we have demonstrated the theoretical and practical viability of creating natural language dialogue systems based upon the four principles of deep representation, uncertain inference, speech act theory, and motivated dialogue control.

The research reported here is merely the beginning of what is sure to be a long-term ex- ploration; the long-term goal of a natural language dialogue system with full human-level functionality remains a signiﬁcant way o. However, the theoretical and practical tools created in the course of this thesis work possess value both as tools for use in real-world software applications, and as demonstrations of what kind of natural language processing it is possible to do using a framework based on the four principles with which we began.

**Keywords：**Dialogue Act; Language Reasoning; Dialogue System

# 

# 目 录

[摘 要 I](#_Toc453623345)

[Abstract II](#_Toc453623346)

[目 录 II](#_Toc453623347)

[Contents IX](#_Toc453623348)

[第一章 绪论 10](#_Toc453623349)

[1.1 研究目的和意义 10](#_Toc453623350)

[1.2 研究现状综述 10](#_Toc453623351)

[1.2.1 知识表示及相关逻辑推理研究综述 2](#_Toc453623352)

[1.2.2 对话管理研究综述 4](#_Toc453623353)

[1.2.3 自然语言理解、生成研究综述 5](#_Toc453623354)

[1.2.4 言语行为理论及相关研究综述 7](#_Toc453623355)

[1.3 存在的问题 7](#_Toc453623356)

[1.4 研究目标和内容 Error! Bookmark not defined.](#_Toc453623357)

[1.5 论文的组织结构 8](#_Toc453623358)

[第二章 智能会话系统中的知识表示和逻辑推理 8](#_Toc453623359)

[2.1 基于超图的知识表示方法 8](#_Toc453623362)

[2.1.1 知识表示和记忆空间 8](#_Toc453623363)

[2.1.2 基于超图的知识表示库 AtomSpace 9](#_Toc453623364)

[2.2 语言和超图之间的转换之范畴论观 10](#_Toc453623365)

[2.2.1 范畴论的基础知识 10](#_Toc453623366)

[2.2.2 语言与超图之间的转换之范畴论观 10](#_Toc453623367)

[2.3 概率逻辑网 10](#_Toc453623368)

[2.3.1 PLN的一个简单概述 10](#_Toc453623369)

[2.3.2 前向和后向链接推理 10](#_Toc453623370)

[2.3.3 一阶概率逻辑网络 10](#_Toc453623371)

[2.3.4 PLN 真值 10](#_Toc453623372)

[2.3.5 PLN 规则和方程 10](#_Toc453623373)

[2.4 本章小结 10](#_Toc453623374)

[第三章 智能对话系统中的情感计算机制 10](#_Toc453623375)

[3.1 Psi模型中的动机行为 10](#_Toc453623377)

[3.2 Psi 模型中的情感与个性 10](#_Toc453623378)

[3.3 本章小结 Error! Bookmark not defined.](#_Toc453623379)

[第四章 基于言语行为理论和概率逻辑推理的智能对话系统建模 10](#_Toc453623380)

[4.1 智能对话系统的概念模型 10](#_Toc453623382)

[4.2 情感驱动的智能对话 Error! Bookmark not defined.](#_Toc453623383)

[4.2.1 OpenCog中的行为选择机制 **Error! Bookmark not defined.**](#_Toc453623384)

[4.2.2 Psi 在 OpenCog 中的使用 **Error! Bookmark not defined.**](#_Toc453623385)

[4.2.3 OpenCog 中的目标 **Error! Bookmark not defined.**](#_Toc453623386)

[4.2.4 执行的管理 **Error! Bookmark not defined.**](#_Toc453623387)

[4.2.5 针对对话控制的OpenPsi的配置 **Error! Bookmark not defined.**](#_Toc453623388)

[4.3 言语规划器 10](#_Toc453623389)

[4.3.1 言语行为自动分类 **Error! Bookmark not defined.**](#_Toc453623390)

[4.3.2 言语行为规划器 10](#_Toc453623391)

[4.3.3 言语行为规划器及其关联的目标和上下文的实现 10](#_Toc453623392)

[4.3.4 实质回应 10](#_Toc453623393)

[4.3.5 问句 10](#_Toc453623394)

[4.3.6 陈述Statements 10](#_Toc453623395)

[4.3.7 自描述Self-Descriptions 10](#_Toc453623396)

[4.3.8 后设话语Meta-Utterances 10](#_Toc453623397)

[4.3.9 命令句、建议Imperatives/Suggestions 10](#_Toc453623398)

[4.4 智能对话系统的概念模型的一种实现方法 10](#_Toc453623399)

[4.5 本章小结 10](#_Toc453623400)

[第五章 自然语言理解：从语言到逻辑 10](#_Toc453623401)

[5.1 链语法 10](#_Toc453623403)

[5.1.1 链语法与短语结构语法 10](#_Toc453623404)

[5.1.2 识别句子的中心词 10](#_Toc453623405)

[5.1.3 RelEx 10](#_Toc453623406)

[5.1.4 RelEx 的系统框架 10](#_Toc453623407)

[5.1.5 RelEx 关系的形式化 10](#_Toc453623408)

[5.2 基于涉身的指代消解 10](#_Toc453623409)

[5.3 RelEx2Logic：逻辑关系抽取 10](#_Toc453623410)

[5.3.1 RelEx2Logic 10](#_Toc453623411)

[5.3.2 RelEx2Logic 所使用的规则 10](#_Toc453623412)

[5.3.3 RelEx2Logic的后处理过程 10](#_Toc453623413)

[5.4 实验结果及分析 10](#_Toc453623414)

[5.5 比较级的处理：实例分析 10](#_Toc453623415)

[5.6 在简单句子集的性能评价 10](#_Toc453623416)

[5.7 本章小结 10](#_Toc453623417)

[第六章 自然语言生成：从逻辑到语言 10](#_Toc453623418)

[6.1 微观规划器 10](#_Toc453623420)

[6.1.1 组块 10](#_Toc453623421)

[6.1.2 指代的引入 10](#_Toc453623422)

[6.1.3 存在问题和改进方向 10](#_Toc453623423)

[6.2 表层生成器SuReal 10](#_Toc453623424)

[6.2.1 SuReal算法 10](#_Toc453623425)

[6.2.2 综观SuReal及其存在的问题和改进方向 10](#_Toc453623426)

[6.2.3 实验结果示例及简要分析 10](#_Toc453623427)

[6.3 本章小节 10](#_Toc453623428)

[第七章 基于超图表示的语言逻辑推理的设计与实现 10](#_Toc453623429)

[7.1 有关比较级的推理 10](#_Toc453623431)

[7.2 基于自然语言的三段论推理 10](#_Toc453623432)

[7.3 本章小结 10](#_Toc453623433)

[第八章 基于概率逻辑推理的问答系统的设计和实现 10](#_Toc453623434)

[8.1 基于超图模糊匹配的问答系统 10](#_Toc453623436)

[8.1.1 基于超图的模式匹配器 10](#_Toc453623437)

[8.1.2 基于超图模糊匹配的问答系统 10](#_Toc453623438)

[8.2 基于后向链接推理的问答系统 10](#_Toc453623439)

[8.3 一个综合的问答规划器 10](#_Toc453623440)

[8.4 实例分析 10](#_Toc453623441)

[8.5 深层语义解析的挑战 10](#_Toc453623442)

[8.6 本章小结 10](#_Toc453623443)

[第九章 总结与展望 10](#_Toc453623444)

[9.1 本文的主要贡献和创新 10](#_Toc453623446)

[9.2 下一步研究方向 10](#_Toc453623447)

[参考文献 10](#_Toc453623448)

[博士期间发表的论文 10](#_Toc453623449)

[致 谢 10](#_Toc453623450)

# Contents

摘 要 I

Abstract II

目 录 II

Contents IX

第一章 绪论 10

1.1 研究目的和意义 10

1.2 研究现状综述 10

1.2.1 知识表示及相关逻辑推理研究综述 2

1.2.2 对话管理研究综述 4

1.2.3 自然语言理解、生成研究综述 5

1.2.4 言语行为理论及相关研究综述 7

1.3 存在的问题 7

1.4 研究目标和内容 Error! Bookmark not defined.

1.5 论文的组织结构 8

第二章 智能会话系统中的知识表示和逻辑推理 8

2.1 基于超图的知识表示方法 8

2.1.1 知识表示和记忆空间 8

2.1.2 基于超图的知识表示库 AtomSpace 9

2.2 语言和超图之间的转换之范畴论观 10

2.2.1 范畴论的基础知识 10

2.2.2 语言与超图之间的转换之范畴论观 10

2.3 概率逻辑网 10

2.3.1 PLN的一个简单概述 10

2.3.2 前向和后向链接推理 10

2.3.3 一阶概率逻辑网络 10

2.3.4 PLN 真值 10

2.3.5 PLN 规则和方程 10

2.4 本章小结 10

第三章 智能对话系统中的情感计算机制 10

3.1 Psi模型中的动机行为 10

3.2 Psi 模型中的情感与个性 10

3.3 本章小结 Error! Bookmark not defined.

第四章 基于言语行为理论和概率逻辑推理的智能对话系统建模 10

4.1 智能对话系统的概念模型 10

4.2 情感驱动的智能对话 Error! Bookmark not defined.

4.2.1 OpenCog中的行为选择机制 **Error! Bookmark not defined.**

4.2.2 Psi 在 OpenCog 中的使用 **Error! Bookmark not defined.**

4.2.3 OpenCog 中的目标 **Error! Bookmark not defined.**

4.2.4 执行的管理 **Error! Bookmark not defined.**

4.2.5 针对对话控制的OpenPsi的配置 **Error! Bookmark not defined.**

4.3 言语规划器 10

4.3.1 言语行为自动分类 **Error! Bookmark not defined.**

4.3.2 言语行为规划器 10

4.3.3 言语行为规划器及其关联的目标和上下文的实现 10

4.3.4 实质回应 10

4.3.5 问句 10

4.3.6 陈述Statements 10

4.3.7 自描述Self-Descriptions 10

4.3.8 后设话语Meta-Utterances 10

4.3.9 命令句、建议Imperatives/Suggestions 10

4.4 智能对话系统的概念模型的一种实现方法 10

4.5 本章小结 10

第五章 自然语言理解：从语言到逻辑 10

5.1 链语法 10

5.1.1 链语法与短语结构语法 10

5.1.2 识别句子的中心词 10

5.1.3 RelEx 10

5.1.4 RelEx 的系统框架 10

5.1.5 RelEx 关系的形式化 10

5.2 基于涉身的指代消解 10

5.3 RelEx2Logic：逻辑关系抽取 10

5.3.1 RelEx2Logic 10

5.3.2 RelEx2Logic 所使用的规则 10

5.3.3 RelEx2Logic的后处理过程 10

5.4 实验结果及分析 10

5.5 比较级的处理：实例分析 10

5.6 在简单句子集的性能评价 10

5.7 本章小结 10

第六章 自然语言生成：从逻辑到语言 10

6.1 微观规划器 10

6.1.1 组块 10

6.1.2 指代的引入 10

6.1.3 存在问题和改进方向 10

6.2 表层生成器SuReal 10

6.2.1 SuReal算法 10

6.2.2 综观SuReal及其存在的问题和改进方向 10

6.2.3 实验结果示例及简要分析 10

6.3 本章小节 10

第七章 基于超图表示的语言逻辑推理的设计与实现 10

7.1 有关比较级的推理 10

7.2 基于自然语言的三段论推理 10

7.3 本章小结 10

第八章 基于概率逻辑推理的问答系统的设计和实现 10

8.1 基于超图模糊匹配的问答系统 10

8.1.1 基于超图的模式匹配器 10

8.1.2 基于超图模糊匹配的问答系统 10

8.2 基于后向链接推理的问答系统 10

8.3 一个综合的问答规划器 10

8.4 实例分析 10

8.5 深层语义解析的挑战 10

8.6 本章小结 10

第九章 总结与展望 10

9.1 本文的主要贡献和创新 10

9.2 下一步研究方向 10

参考文献 10

博士期间发表的论文 10

致 谢 10

# 

第一章 绪论

从人工智能发展的早期开始，实现人机之间的自然语言交互一直是该领域的研究焦点，1950年，Alan Turing提出了一个如何判定机器是否具有智能的标准“图灵测试”。至此，对话系统更是成为判定机器是否智能的理想模型。然而，对话系统一直是自然语言处理和人工智能领域的研究难题，它不仅涉及到对自然语言的处理，自然的对话系统还要求机器具备一定的认知能力，使其能按照人的响应方式与人类进行交互。本文从目前对话系统研究中亟待解决的认知技术角度出发，对此展开深入研究。

本章简要介绍认知对话系统的研究背景和意义、以及相关的研究现状和存在的问题，最后列出本文的主要研究内容和组织结构。

## 研究目的和意义

与机器进行自然的智能对话一直是人们梦寐以求的梦想。随着人工智能技术的不断发展，对话系统逐渐成为人工智能在商业领域的焦点，各大公司也都纷纷高调发布对话系统产品，如在Jeopardy节目中夺冠的IBM Waston，苹果的Siri，微软的Cortana和小冰，小i和图灵等等。尽管近年来，对话系统相关的计算语言学和人工智能领域的很多技术都取得了巨大的进展，如语音识别和合成、机器翻译、信息检索、浅层语义分析甚至情感分析等，但对话系统的认知对话管理和语义语用等深层理解领域仍没有根本性的突破。即使是目前表现最好的对话系统，也依然是基于表层的语言分析和简单的启发性知识来实现。然而如果对话系统无法作为一个认知主体，以拟人的方式去感知和交互，就很难激发人类与其长时间交流的兴趣。

对话系统涉及的领域很广，本文重点考虑认知型的对话系统，也就是强调机器作为一个认知主体，去理解和适应人的交流方式，从而使用户能用与人交流的自然方式去和机器交流。本文中提到的对话系统中的相关认知技术主要包括知识的深层表示、不确定性逻辑推理、语言与涉身知识的融合以及动机驱动的对话控制。对这些领域的研究显然具有重要的研究价值。这些领域中任一组块取得进展，不仅仅对于认知对话系统，而且对于其他领域，如语义理解、情感建模、自动化学习系统等等，都会有相当大的辅助和促进作用。另一方面，对话系统中的认知技术得到改进，无疑能大幅度提高对话系统的实用价值。能帮助机器更好更准确地结合语境理解和适应用户的交流方式，从而更准确更人性化地帮助人类完成一系列的辅助工作，拟人的交流方式也会使人们觉得更舒心和信赖，这也将为人们的生活带来诸多便利。

尽管拥有巨大的研究价值和市场价值，研究者也发现构建一个能达到人类水平的对话系统是一项十分艰巨的任务。本文的研究着重于对话系统中的相关认知技术，提出了一个动机驱动的对话控制模型，即结合了心理学上的动机驱动模型，言语行为理论，以及基于概率逻辑网络的超图形式的知识表示，赋予对话系统一定的认知能力，从而使机器能作为一个认知主体，以更自然的方式与用户进行交互。

## 研究现状综述

对话系统简单来说就是计算机用来试图和人类通过自然语言交流的软件系统。对话系统的思想可以追溯到1950年图灵在[1]中提出的图灵测试，在过去的几十年中，有关对话系统的研究大致有两个不同的方向：一个是仅在表面外观上模拟对话，也被称为聊天机器人，如著名的聊天机器人心理医生ELIZA[2]等。这类系统大多数不考虑理解对话内容的含义，直接采用模板匹配的方式找到和用户输入相关的对话，目前经常被用在各类社交网站和实时通讯的社交软件中。另一个是试图模拟人类真实的对话，并动态产生合适的对话，也被称为对话管理。后者是本文的研究重点，因此这一小节我们将回顾这一类对话系统以及对话管理的相关研究。

目前，对话系统的研究中有很多不同的架构体系，一个对话系统包含哪些功能模块也视不同的系统而异，但一般来说，对话系统主要包括以下几个功能模块：自然语言理解模块、对话管理模块、知识表示模块、自然语言生成模块[3]。各模块之间的数据走向如图1.1所示。接下来我们将分别就对话系统中的各个功能模块来展开综述相关研究现状。另外，由于本文的认知对话系统中使用了言语行为理论的启发和指导，因此也会在本节对其相关研究进行简明的阐述。需要说明的是，本文研究的对话系统暂时只是基于文本的，所以这里暂且不讨论语音识别和语音合成模块。

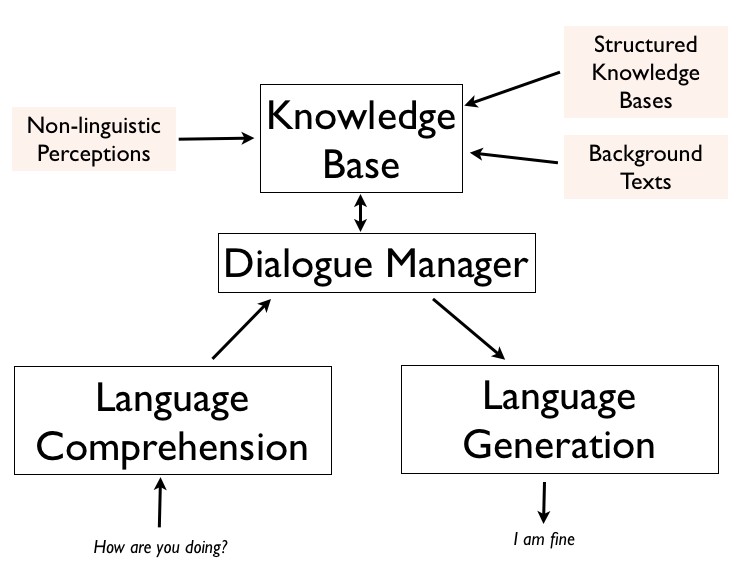


图 1.1 对话系统的基本结构

### 知识表示及相关逻辑推理研究综述

对于任何以实现复杂功能为目标的自然语言处理系统（如对话系统或复杂的问答系统）来说，该系统内部的知识表示形式，即如何表示语义处理后的自然语言，往往是系统的各方面性能表现的关键所在。本小节将介绍知识表示目前的研究现状并回顾一些典型的知识表示方法及相应的优缺点，我们也会在本节重点介绍与本文所采用的知识表示相关联的几个具有代表性的知识库。

如何合理有效地表示自然语言话语的含义是过去50年来一直被争议的问题，这也是从一个句子中抽取信息的语义建模问题所面临的第一步。然而目前，“知识表示”需要满足哪些条件仍然处于争论状态。Jurasky和Martin在经典的《自然语言处理综论》[4]的第14章中从不同方面总结了一个完整的知识表示体系所要满足的计算需求，其中包括：

* + - 可能性验证(Veriﬁcability)：必须能够确定知识表示的真实性，即知识表示体系能将输入的命题表示与储存相关知识的世界知识库中的表示相比较或匹配。
    - 无歧义表示(Unambiguous Representation)：知识表示的语言支持的表示只能有一个无歧义的解释。
    - 规范形式(Canonical Form)：表达同样信息的输入必须具有同样的知识表示形式，即规范形式。
    - 推论与变元(Inference and Variables)：知识表示体系能根据输入的知识表示以及储存的背景知识推理出可靠结论。
    - 表达能力(Expressiveness)：知识表示方法需要具备足够的表达能力来处理各种广泛领域的知识。

直到最近，这些需求列表一直被当做是知识表示领域中一个相当完善的黄金标准。Yarin等人[5]指出了这个必要条件列表并不完整，并指出了知识表示体系还需满足下列需求：

* + - 不确定性和内涵的表示
    - 能使用可行的方式来抽取话语的信息（即，从数据中抽取相关知识表示的算法的需求）
    - 从话语中推理出相对应的情感和观点
    - 从不同命题中获取相关的不确定性
    - 不仅需要区分具有不同的意义的话语，还需区分涉及相似意义的话语
    - 在从话语中获取活动信息的同时还要获取活动相关的动作者(agent)
    - 回答问题除了使用相关的知识外，还需要能利用这些知识和问题的意义。
    - 对同一个意义，能生成不同表达方式的话语

当然这些需求本身也没有得到相关的验证是否就完全，但近几年提出的一些知识表示方法及相关研究也在试图满足这其中的部分或所有需求。下面我们将简单回顾一些具有代表性的知识表示方法以及相应的优缺点，从中也不难看出未来知识表示领域的研究方向。知识表示的任务比较繁琐，所涉及的领域比较广，因此很难有一个完整的分类体系来概括所有的知识表示方法，本文通过以下几方面来分类介绍典型的具有实用价值的知识表示方法。

1. 传统的知识表示方法

这里的传统的知识表示方法主要指逻辑表示法、框架表示法以及语义网络表示法，以及在这基础上的一些延伸表示方法，下面将对这几类典型的传统知识表示方法做简单的分析和对比。

1. 逻辑表示法

逻辑表示法的思路是将自然语言话语转换成某种逻辑约束下的一组逻辑表达式，以及在该逻辑上对这些逻辑表达式的语义解释。常见的用于知识表示的逻辑有命题逻辑，传统逻辑以及谓词逻辑等。对复杂的自然语言来说，命题逻辑和传统逻辑的表达能力相对弱，这里主要介绍基于谓词逻辑的知识表示方法。谓词逻辑表示法中既包括一阶逻辑也包括高阶逻辑（如多个量词）的表示，但从实用角度来看，也就是在有限数量信息的下，两者之间可以通过一定的逻辑等价推理来转换，而一阶逻辑具有简洁和易推理的优点，因此相对来说更广泛地被应用，也是被研究地相对成熟的一种知识表示方法。基于一阶逻辑的知识表示体系中使用了三种基本构件：常量（有时也称为原子atom），函数以及变量，每一个构件都指向世界知识中的某个对象。其中，常量通常用于表示世界知识中的对象，比如“狗”；函数通常定义在常量或者变量上，在自然语言的背景下，“函数”可用于产生新的词汇。比如函数“Black(狗)”可生成“黑色的狗”或者“黑狗”；变量则允许我们表达对非具体对象的陈述或预测。将这三个简单的构件和存在量词以及连接符号和否定关系符号，我们可得出以下的表达式用于表示“存在一个实体，这个实体拥有一只黑色的狗，并且给这只黑色的狗喂了红色的肉。”

∃*X*(*Own*(*X*, *Black*(*Dog*)) ∧ *Feed*(*X*, *Black*(*Dog*), *Red*(*Meat*))) (1-1)

在上述的逻辑表达式上，我们可以进行相关的逻辑推理和蕴含等操作，并且能通过。一阶逻辑表示法允许我们将自然语言话语表示成一种无歧义的表示方式。对于多义词，我们可以采用多个常量（原子）来表达，比如可以用Bankinstitution来表示意思为“银行”的”Bank”，用Bankinstitution来表示意思为“河岸”的”Bank”。此外，这种表示方法的表达能力也适用于表达多个句子的情况。

比较典型的使用一阶逻辑的知识表示体系有早期的Cyc系统[6]。Cyc系统拥有一个非常丰富的人工编码知识库，它的终极目标是：以谓词逻辑的形式，将所有人类常识进行编码。虽然它的知识库中已存有数百万的谓词逻辑表达式，但到目前为止，似乎只对一小部分人类常识进行了编码。

一阶逻辑的知识表示方法相对简洁，但由于自然语言的复杂性和不确定性，如何将自然语言话语转换成一阶逻辑的表示形式，是一个极大的挑战。Pianatadosi等人[7]首先在几十个词的限定条件下构建一个概率模型，使其能对各种可能的语法建模并过滤掉过于复杂的，从而根据这些语法模型来得到相应的一阶逻辑表示形式。在此基础上，他们的研究还深入到通过递归方式来遍历所有可能的上下文无关文法的概率模型，使得递归进行到一定程度能得到一个对话语进行最佳解析的一阶逻辑表示形式。该方法最终能得到一个不可扩展的在迷你规模语法基础上的组合语义获取模型，并没有实用价值。

后期的Cyc系统也为了克服一阶逻辑表示方法的缺点，引入了贝叶斯网络来为该知识库中的概念和事件提供相应的概率真值，但后期Cyc系统也转入商业用途，其数据和表示也不免费公开，因此我们尚不清楚目前其具体研究状况。

1. 基于框架的知识表示法

基于框架的知识表示方法又被称为槽填充表示法。框架(Frame)可以理解成一种将某一对象或事件所涉及的知识信息存储在一起的复杂的数据结构，通常包括主体和槽(Slot)两部分，其中主体是固定的，表示一个特定的概念、对象或事件，而槽用于表示主体的各属性及属性值。其一般形式可以表示如下：

<主体>

<槽名 1><槽值 1>| <侧面名 11><侧面值 111，侧面值 112，……>

<侧面名 12><侧面值 121，侧面值 122，……>

比如“我吃蛋炒饭”可以表示成框架形式如下：

<吃>

<施事者><我>

<受事者><蛋炒饭>

基于框架的知识表示常被用于语义角色标注中，即标注出句子中的谓词以及相应的参数。比如“我吃蛋炒饭”，语义角色标注可以标注出其中谓词是“吃”（框架主体，表示事件），该事件的槽值可以表示为施事者“我”，受事者“蛋炒饭”。

基于框架的表示法具有代表性的系统有FrameNet[8]，FrameNet是一个人工编写的基于框架的语义知识库，其中包含了1,200个语义框架，13,000个词汇单元，大约19万个例句，目前被广泛用于语义角色标注的训练语料。基于框架的表示形式也可以很容易被转换成等价的一阶逻辑表示形式。与一阶逻辑表示相似，将自然语言话语转换成基于框架的表示形式也是该表示方法的巨大挑战。

1. DSC树的表示方法

鉴于将自然语言话语转换成上述两种表示形式非常困难，Liang[[9](#_bookmark125)] 等人试图将该任务简化成一个允许进行高效推理的约束满足的问题。为此，他们提出了一种新的知识表示方式：基于依存的组合语义树(Dependency based Compositional Semantics Tree，以下简称DCS树)，并通过预先设定的触发词和谓词的约束来高效地将自然语言转换成DCS树的表示形式，如单词city将会触发相应的谓词city。DCS树通过树的形式来构建逻辑表达式，它与依存句法树平行，为自然语言的语义解析提供了很多便捷。尽管该方法的提出只针对小规模数据集合上和预先设定的触发词表，但不难通过语义角色标注和浅层语义分析抽取出语句中的谓词等自动方法来代替这些人工编写的词汇表，从而扩大该方法的应用规模。此外，该表示方法还能结合大规模的事实数据库，如Freebase[[10](#_bookmark125)] 或者Google的知识图表，可以通过简单的语义角色标注和基于对先前观察的语句来填充语义空槽，从而实现对现有问答系统的扩展。

尽管DCS树的表达方式在有限领域和数据集合上能达到较好的效果，但从其研究[[11](#_bookmark125)] 来看，该表示方法的表达能力仍然是个悬而未决的问题，目前也没有任何相关研究表明其表达能力与一阶谓词逻辑的表示方法相当。从另一个角度来看，DCS树的表达方式受数据库查询语言的启发[[5](#_bookmark125)] ，通过数据库查询语言来表示自然语言语句的语义为知识表示提供了一个全新的思路，并不仅因为它将数据库查询语言的成熟研究也应用到语义表示和学习方面，而且它也实现了在约束满足条件下有效进行逻辑推理和蕴涵[[13-15](#_bookmark125)] 。然而即使数据库查询语言相当成熟，但其表达能力仍然是非常有限的，因为这些语言除了无法表示聚合函数和相关的算术运算，还无法表达递归查询[[12](#_bookmark125)] ，但递归结构在自然语言中显然是相当常见的，这在一定程度上间接说明了DCS树的表达能力不一定足够用于实用的自然语言处理应用领域。

1. 语义网络表示法

语义网络的知识表示是人工智能和自然语言处理等领域的又一种经典的知识表示方式，它主要用于表示陈述性知识。语义网络在形式上是一个加权有向图，其中有向图的结点表示某个概念、对象、事件或状态等，边（也称为弧）则表示结点之间的语义关系。

语义网络中的语义关系通常包括实例关系（ISA，用于表示实体结点和类结点之间的关系，如“史努比”和“狗”）、泛化关系（AKO，用于表示类结点与抽象层次更高的类结点，如“鲸鱼”和“哺乳动物”）、部分整体关系（Part-of，用于表示结点与其组成成分之间的关系）以及属性值关系（用于表示个体的各属性和对应的属性值之间的关系，通常使用语义网络中的边来表示属性，用边所指向的结点来表示该属性的值）。除此之外，语义网络还能表示以谓词或关系为中心组织语义关系，还可表示不同子网络之间的逻辑关系，如析取、合取、否定和蕴涵等。

语义网络的知识表示体系所采用的推理机制主要有匹配和继承，继承一般通过匹配和搜索来实现。在问题求解时，可以根据问题的描述来构造一个语义网络片段，并在语义网络知识库中查找可以与该语义网络片段匹配的子网络，然后根据查找到的匹配来回答问题。

目前被广泛使用的基于语义网络的知识表示库有ConceptNet[[16](#_bookmark125)] 和谷歌的知识图谱等。语义网络非常符合人类的思维习惯，其表达方式自然、简洁，易于理解。表达能力方面，语义网络不仅可以表示事物的属性状态、行为动作、目标功能等，而且还能表示事物之间的关联，为人工智能和自然语言处理等相关系统提供了便捷的知识推理平台，但是语义网络也面临着诸多挑战，首先，语义网络的形式过于简单，很难表达相对复杂的关系类型，复杂关系的引入很容易增加语义网络的复杂度，从而使知识的储存和检索过程变得相当复杂，甚至难以实现。其次，语义网络中的语义关系查询和推理往往需要计算复杂度非常高的图算法，而且基于图的算法一般可扩展性差，这就直接导致，在知识库达到一定规模后，将面临着计算效率问题的挑战。最后，该表示方法还面临着严重的数据稀疏问题的挑战，对那些在语义网络中对外连接较少的实体，意味着涉及到它们的路径很少，这导致图算法对其相关路径查询和逻辑推理带来很大的难度。

上述的几种表示方法都可以看成是刚性表示方法，因为它们都无法在逻辑推理和蕴涵过程中处理非确定性。然而真实世界中大部分知识都是非确定性的，近年来也有将概率引入到这些传统的知识表示方法中，如[[18](#_bookmark125)] 中将依存树转换成准概率逻辑形式，并使用了马尔科夫逻辑网络[[19](#_bookmark125)] 进行语义解析，这样一来，该方法既能进行逻辑推理和蕴涵，同时处理其中的不确定性。然而，该表示仅限于表达一些基本形式的语句，[[20](#_bookmark125)] 将上述工作中的表示继续简化，在马尔科夫逻辑网络中使用了Pitman-Yor随机过程来对谓词和变量之间的概率依存关系进行建模，使其更适用于简单的角色标注的形式。这些仅仅是在传统逻辑表示上引入了概率随机过程等，下面我们将简单介绍一些基于概率逻辑的知识表示方法。

1. 基于概率逻辑的知识表示法部分

随着大数据技术的不断发展，不确定性的知识表示和管理在人工智能各领域越来越被重视起来，如上所述，基于概率逻辑的知识表示使得知识表示库能以认知概率形式来储存世界知识中的不确定性。因此，近年来，基于概率逻辑形式的知识表示方法不断涌现并广泛应用于各类学习系统中，如马尔科夫逻辑网（MLN），归纳逻辑编程（ILP），概率逻辑编程（PLP），贝叶斯逻辑编程（BLP），随机逻辑编程（SLP），随机关系模型（SRM），概率关系模型（PRM）等等。篇幅关系，我们这里只简单介绍和本文所使用的知识表示方法相似的几个典型的基于概率逻辑的表示方法。

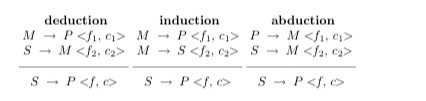
贝叶斯逻辑编程 (Bayesian Logic Programs，以下称BLP) [[21](#_bookmark125)] 采用贝叶斯网络来表示知识，其中节点表示命题子句，它是众多“概率Prolog”中的一种。BLP可以用于表示任意带概率的Prolog形式的关系，并能通过逻辑推理来传播这些概率。BLP中维持了概率层次上的“直接因果影响”和逻辑层次上“间接推理”的同构关系，使得该结构中的概率和逻辑推理能达到一致，而无需借助棘手的混合模型。Puech[123]对BLP的表达能力进行深入研究，并将其与SLP（Stochastic Logic Programs）的表达能力进行对比，其结果显示SLP能将同样的知识表示成BLP的一个子类。但BLP中的推理机制（即通过结构化的EM算法来查找BLP结构）却比SLP要简单的多，而SLP中的推理机制是被公认为非常困难的[[22](#_bookmark125)] 。然而，由于BLP依赖于Prolog，也继承了Prolog的不足之处，如依赖于特定的逻辑表达式归一化、缺少可扩展的推理机制等。

马尔科夫逻辑网（Markov Logic Network，以下称MLN）最初由Domingos等人提出[[19](#_bookmark125)] ，MLN采用一阶谓词逻辑来表示实体之间的逻辑关系，采用马尔科夫网络形式的概率图模型来推理其中的概率分布。其基本思想是使一阶谓词逻辑中的硬性规则约束有所松弛，换句话说，当一个可能世界违反了其中某条规则，这个可能世界存在的概率很低，而并非直接设为0。MLN对每条规则都设定一定的权值，用来表示这条规则对可能世界的约束力，当一个规则的权值很大，那么违反该规则的可能世界存在的可能性就越低。MLN较好地结合了一阶逻辑和概率图模型的复杂性和不确定性表示，因此在该领域曾被广泛应用，如[124-126]，其中也包括在NLP的一些领域上的应用，如实体抽取和语义关系抽取等。然而，由于在概率图模型中的推理复杂性，MLN近几年来慢慢失去了热度[127] 。MLN假设整个网络都是基于一个概率分布的，这显然也不太合常理，而本文采用的PLN以及下面要介绍的NARS[[24](#_bookmark125)] 就刻意避开这样的假设。

与传统的知识表示方法相比，上述的知识表示方法的有点在于能在逻辑中表示认知的不确定性，然而，这些这些表示结构都是离散的，无法用来推理句子之间的相似度。因此[[22](#_bookmark125)] 提出了PLP（Probabilistic Similarity Logic ）来引入原子之间的距离衡量机制，但目前只能用于简单的不确定性逻辑推理中。本文的认知对话系统中所使用的概率逻辑网（Probabilistic Logic Network，以下称PLN）也有类似的地方，且能够将其他非语言渠道（如机器视觉感知）获得的相似度通过概率推理整合得到句子之间的相似度，并整合到其他的逻辑关系（如继承关系）中。

Pei的NARS[[24](#_bookmark125)] 引擎运用了传统逻辑推理法，并引入了独特的数学理论来管理传统逻辑关系中的不确定性。但NARS是构建在经验的基础上，而不是基于模型轮语义。 在许多方面，我们所使用的PLN逻辑形式化体系与NARS有类似的地方，但也存在巨大差异。PLN在一套独特的数学理论基础上同时采用了传统逻辑和谓词逻辑，并且根据概率论和模糊数学理论来推导出不确定性真值的公式，而NARS的不确定性推理体系则基于原始的非逻辑形式。

图1.2中展示了基本演绎推理、归纳推理和外展推理公式。这些公式是PLN和NARS共有的。在每个关系式右边的<s,c>表示“每个关系的强度和置信度”。PLN和NARS使用不同的公式，从那些前提中推导（优势、信息）结论的真值。有关PLN的知识表示和推理机制我们将在第2.3节中进一步介绍。

图**1.2 NARS/PLN** 传统逻辑中演绎推理、归纳推理和回溯推理的形式

1. 组合分布式表示部分

分布式表示方法被用于词汇语义学领域已超过40年[[4](#_bookmark125)] ，该方法对词汇使用连续表示，也就是说，在表示一个词汇时，同时将其相邻的词也表示在二进制向量中，比较典型的例子就是被广泛使用的词袋模型（bag-of-words model）。但是这类表示方法无法表示较复杂的知识，因此逐渐被组合分布式的表示方法所取代。

[128]中提出了将连续分布表示方法和传统的组合表示方法结合起来用于知识表示，并使用张量积（tensor product）将信息从词汇层次带入到更高层次，然后使用线性函数将其映射到较低维度的句子空间中。Grefenstette[[26](#_bookmark125)] 指出该表示方法是可以用于表示非确定性的，还解释了其中的线性函数映射等同于非量化的一阶逻辑；但也证明了量化的一阶逻辑表示是无法通过线性函数映射得到的。因此，他在该文研究中引入了非线性映射用于获取量化的一阶逻辑表示。

组合的分布式知识表示方法还被用于语义建模领域中[129,130] 。它提供精确的表示，具有较好的表达能力，常被用于情感和角色标注方面的推理，而且能表示不确定性和句子的真值，目前该领域的研究主要集中于在句子空间中有效地进行语句的含义抽取和逻辑推理，但仍然有待深入研究。

本文采取了基于超图形式的知识表示Atomspace，并在该表示上使用了概率逻辑网络PLN进行知识推理和学习，这将在第二章中更详细地介绍。

### 对话管理研究综述

对话管理，也称为对话控制，是决定一个对话系统在对话中说什么的关键。一般来说，对话系统首先利用自然语言处理模块将用户的输入转换成系统所使用的知识表示形式，即该系统所能理解的语义形式，而对话管理模块会结合当前的语境、对话历史以及自身的知识库等因素输出一个概念层次上的应答。最终，自然语言生成模块会对这些概念层次上的应答转换成自然语言输出。

对话管理模块在不同的对话系统中完成的任务是不同的，但其主要功能可以归纳为以下几个方面：

* + - 搜索和查询：根据当前的输入以及语篇上下文，在知识数据库中搜索查询和用户输入相关的知识或可能应答内容。
    - 询问：如果无法查询，针对某一问题，询问更多相关信息，直到能提交一个合适有效的查询。
    - 确认：当用户的输入无法被理解时，反复请求确认语焉不详的信息，使得用户输入的信息更具有操作性。
    - 预测：预测对话的进行方向。为对话系统的下一步操作即自然语言生成模块提供概念层次上的应答内容。
    - 控制：为了能实现自然流畅的类似人类的对话，采用一定的对话控制及交互策略，如：介入对话；回应惯用语；多方对话等。

在目前的研究现状下，对话管理模块在几乎所有目前存在的对话系统中起到了关键的作用。在实现方面，对话管理必须找到并返回能使对话保持与对话历史协调的最合适的响应。迄今为止，随着越来越多的相关研究和发展，对话管理的处理方法已经有很多，目前主流的对话管理方法可以分为三类：基于知识的对话管理；数据驱动的对话管理；基于混合方法的对话管理。下面会分别介绍这些对话管理方法的研究现状。

1. 基于知识的对话管理

早期的对话系统都是由具备特定领域知识的开发者来设计，如SUNDIAL[27]以及ARISE[28]。这类系统通常仅限用于完成高度结构化的任务，其中的对话也仅限于特定领域以及规范化的对话。基于知识的对话管理方法一般采用自动有限机来实现，往往涉及很多手工编写的规则，而这些规则通常与应用所需的知识密切相关，而且需要通过用户的不断使用来改进和完善。基于知识的对话管理方法常被用于具有明确结构和目标的强类型对话系统的快速建模，如[29]。该方法也因其简单易实现的特点而被一些实际应用系统采纳，如自动服务热线等。然而，该方法也有很多局限性：第一，由于规则的限定性和对话的灵活性，扩展其中的手工编写规则是非常困难的。第二，其对话流程也是比较刻板的，因为该方法的对话必须按照系统定义的结构化流程来进行。例如，当用户通过系统提出的问题而提供多余的信息时，系统受其限定结构流程限制，将无法处理这些信息而忽略用户提供的补充信息或者无法产生应答。第三，该方法的可移植性很差，其对话规则往往依赖于特定领域的知识，因此无法被用在其他领域甚至相似领域的对话系统上。

为了克服这些局限性，[30,31,32,33]提出并改进了更通用的基于议程或任务的对话建模方法，该方法通过更强大的知识表示方法，将大任务或大议程分解为更小更容易处理的子任务或子议程，而这些子任务和子议程也能使用该通用的对话管理模式。目前最流行的对话管理框架之一RavenClaw[34]中使用的便是基于议程的对话建模，它采用了双层对话管理框架，其中一层用于受限领域的对话控制，另一层用于非受限领域的对话控制。其中受限领域的对话控制通过分层的树结构来实现对话交互，如图1.31.5；而非受限领域的对话任务则可通过使用非受限领域对话管理层来执行给定的对话任务来完成。虽然这种通过双层对话管理框架来实现的基于知识对话管理在一定程度上提高其灵活性和可移植性，但是其仍然摆脱不了知识源的受限，需要相关领域的专家来设计其中的任务分层结构以及议程计划等等，因此实现过程仍然是很费时间的。针对该问题，随着语料库语言学技术的发展，从对话语料库中自动获取相关知识源以及对话结构的方法兴起并发展起来，如[35,36,37,38]。其中[35]使用了无监督聚类算法从通话转录的数据库中抽取相关信息并构建了一个呼叫路由领域的对话管理模型（或称为话题结构）。

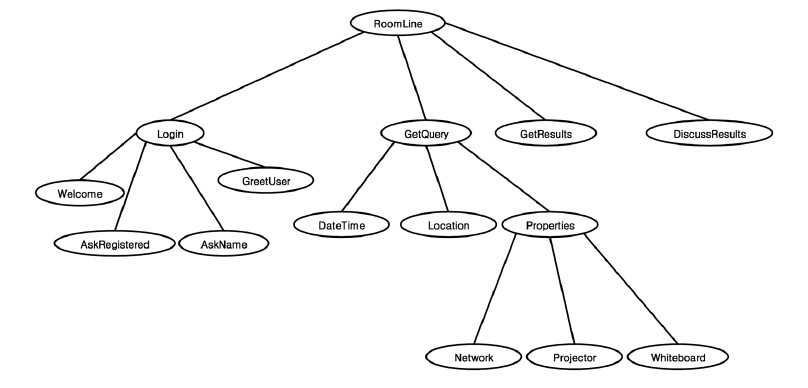


图 1.3 基于 RavenClaw 框架的 RoomLine 对话管理策略树

1. 数据驱动的对话管理

最近几年，对话管理的研究趋势逐步转向数据驱动的方法，即通过语料库来训练相关对话结构和知识库。虽然这种数据驱动的方法需要消耗很大的时间和人力进行语料库标注，但是对话管理模型的训练几乎不需人工干涉，而且在可移植性方面，如果需要在其他领域建立一个对话系统，也只需要重新训练新的语料库，与基于知识的对话管理相比，这显然是更简便的方法。

这些优势驱动了使用无监督强化学习的随机对话建模的发展，如[?]使用了基于马尔可夫决策过程（MDP）的强化学习算法，[39,40,41,42,43]使用了基于部分可观测马尔可夫决策过程（POMDP）的强化学习算法来进行对话建模。这些对话管理框架都使用数据驱动的统计机器学习算法，通过强化学习算法来优化系统的奖励或者惩罚函数，允许系统有理论原则性地根据当前的对话状态来动态修改对话策略。此外，基于POMDP的对话管理还能通过支持最优语音识别和自然语言理解结果来评估信念状态（belief state），从而达到纠正语音识别以及自然语言理解模块中出现的错误。对于基于POMDP的对话管理，系统只观测在当前状态下的有关现实世界（在对话系统中指的是语音识别和自然语言理解结果）的不完整的信息，系统必须利用当前对话状态下寻求一种最优策略将当前的信念状态映射到对话管理策略上。因此，基于POMDP的对话管理可以通过维持信念状态（即最优语音识别和自然语言理解结果）分布来实现语音识别和自然语言理解结果的纠错处理，而无需其他特殊方法。同时，有关用户的背景知识（如用户的特长以及情感状态等）也能通过可分解的强化学习模型映射到状态空间中，从而使得对话管理更人性化。

尽管强化学习算法在理论上能解决基于不确定性的推理和决策，从而为对话管理提供了有效的途径，但在实现过程中，基于强化学习的对话系统依然困难重重[44]，首先，对话系统中的状态包含大量语义项和项值、用户意图、理解结果和对话历史的各种可能组合，使得状态空间的规模指数增长[45]，因此使得强化学习的过程计算复杂度巨大，也使得对话控制的精细改进过程变得困难；其次，强化学习得到的最优策略也剥夺了对话系统开发者对对话流程的监督指导权利。[46,47,48,49,50]在一定程度上对这些问题进行了改进。[47,50]中研究了如何将传统的基于知识的对话管理和基于强化学习的对话管理结合起来构建商务领域的对话管理规则库，在一定程度上给开发人员留下了对对话管理的监督和指导的空间。然而，目前该方法仍在研究中，尚未用到实用性的对话系统中。

数据驱动的对话管理方法还包括使用最大似然估计等有监督的机器学习算法从对话语料库中训练对话模型[51]。为了避免数据稀疏的问题，[51]中采用了对话寄存器来表示对话状态序列用于跟踪对话历史，其中对话寄存器包含用户在整个对话历史中提供的各项信息。

基于实例的对话管理方法也是数据驱动的对话管理中的一个流行的研究方向，具有代表性的系统有[[52,53,54]。该方法假定类似的对话状态会引发类似的回应，因此可以通过匹配对话实例库中的与当前对话状态最相似的对话状态来构建相应的对话模型。目前搜索与当前对话状态最相似的对话实例大多数都使用关键词来检索，对话实例库中的对话实例的表示通常采用语义约束的形式，从而整个对话实例库可以通过语义索引来概括[54]。对话管理模块首先将当前的对话状态也表示为语义约束的形式，并试图从对话实例库中找到与之相近的对话实例，如果没有返回结果，系统会放松语义约束后再一次从对话实例库中查找相近的对话实例；如果返回结果不止一个，那么系统会采用启发式算法来计算返回结果中的每一个实例与当前输入的相似度，然后选择相似度最高的返回作为应答。

数据驱动的对话管理的相关研究在近几年来发展可以说是突飞猛进，当下流行的深度机器学习也被用于对话管理领域，通过深度神经网络在大型对话语料库上训练对话模型的研究也取得了比较有前景的实验结果[55,56,57]。近期的相关研究还包括“端到端”的数据驱动对话管理框架，也就是利用多层次的神经网络对整个对话过程中的每一个组件以及每一个阶段都进行训练，从而得到包含从语音输入到语音输出，以及对话过程中各个阶段的一系列概率模型[58]。然而，这些数据驱动的方法都严重依赖于足够大的数据集，为了取得有价值的实验结果，动辄需要上千万甚至上亿则对话。

1. 基于混合方法的对话管理

为了克服上述提到的基于知识以及数据驱动的对话管理方法中的各种不足，基于混合方法的对话管理的研究逐渐发展起来。

传统的基于强化学习的对话管理方法需要通过用户和系统的反复交互来学习一个好的对话策略，而且目前的强化学习算法的泛化能力很弱，往往需要通过大量人力去和系统交互或者庞大规模的对话语料库去归纳一个很小语境条件下的对话策略。为了解决该问题，[59,60]中引入了用户模拟器的技术，也就是说，通过用户模拟器来代替人和对话管理器进行交互。根据有限的真实语料概括学习得到特定的用户模型，用户模拟器可产生大量的模拟对话。除此之外，[61]中还提出了将强化学习和有监督学习结合起来并在有限固定的对话语料库上学习和优化对话策略的混合方法，有监督学习的引入在一定程度降低了传统强化学习方法中对巨大对话语料库的需求程度。在该方法中，强化学习用于优化对话策略中对话奖惩的度量，而有监督学习用于指导对学习到的对话策略状态空间的剪枝，从而得到更高效有用的状态空间。

在传统的POMDP算法中，优化过程可以在任何时刻选择任意的行为，因此，没有简洁易操作的方式引入一些领域相关的常识性知识来约束和指导优化过程。例如，一个医疗相关的对话系统不应该在问病人症状之前就提供用药建议，然而，传统的POMDP算法却不存在直接的方法将该常识引入到优化过程中。近年来，也有相关研究提出来改进该问题，如[47,50]中提出了在POMDP框架中基于传统规则来约束策略状态中的可能对话行为的集合，这样通过传统规则来对状态空间中的一些不合理的行为进行剪枝，不但加速了强化学习的优化过程，也使优化过程往更可靠的方向进行。

目前，传统的基于实例的对话管理方法应用在实用口语对话系统中还存在很多关键问题，比如先验知识的缺乏，语音识别的错误，以及语义解析的不确定性等。针对这些问题，[62]中提出了将对话实例与先验知识同时用在基于实例的对话管理中，从而提高系统的鲁棒性。该方法使用了基于议程的模型，将先验知识表示为议程图，并作为对话管理的子任务流之一来对领域相关的对话控制进行编码。这些先验知识不仅被用于预测系统的下一个对话行为，还通过议程图来跟踪对话状态，以便于处理一些话题焦点转移的意外情况。另外，该方法还通过在话语层次和语篇层次上的打分作为启发，来对当前对话状态下的N-best的对话行为假设进行重排。

基于混合方法的对话管理的最新研究趋势还包括在强化学习模型中引入概率规则的概念，[63]中将概率规则定义为逻辑条件和概率事件之间的结构映射，并作为高层次的模板用于概率图模型中，其中可能包含未知参数，其值可以通过数据进行贝叶斯推断来估计得出。由于使用了逻辑的抽象化，这些概率规则可以对数据驱动的基于POMDP的对话管理中使用的概率模型进行编码成更紧凑更可读的形式。在理想情况下他们需要较少的参数估计，因此，与典型的基于POMDP的对话管理方法相比，要求较少的训练数据，也允许开发者在对话管理过程中使用人工编写的对话控制规则。然而，该方法还未成熟和完善，其实用价值在很大程度上还有待考察。

在这些繁多的对话管理研究技术中，本文提出的方法更接近于现有技术中的通用对话建模技术，以及最新的在对话管理中引入概率规则的相关研究。然而，与这些研究又不同，本文由系统的动机出发，结合言语行为以及其他的交际行为与这些动机之间的关系来构建一个相当复杂的对话管理模型。此外，由于使用了基于概率逻辑规则的知识表示方法，本文的认知对话管理框架也能通过观测数据来进行对话模式的自适应。因此，本文的方法可以被认为是基于混合方法的对话管理。从长远角度来看，本文提出的对话管理框架将实现以通过观测数据来进行经验概率分析为主的对话控制，不过目前，我们的研究集中于将其内置的通用对话建模将作为对话行为控制的主导。

### 自然语言理解、生成研究综述

认知对话系统的研究、开发和使用都离不开自然语言理解和自然语言生成的支持，因此本小节将分别对自然语言理解和自然语言生成方面的现有研究技术以及存在的问题等进行简要的回顾。

1. 自然语言理解

自然语言理解是自然语言处理（NLP）的分支领域，它是通过使用软件来理解自然语言的意义（语音，或者本文更关注的“文本”）。自然语言理解主要是将自然语言转换成抽象的语义形式，以获取语言的含义；或者是转换为某种响应（例如对某个问题的回答），以说明其理解了语言的含义。

在20世纪80年代之前，自然语言理解的方法通常是基于人工编写的复杂规则。紧随着机器学习技术的发展，基于统计的自然语言理解方法也开始兴起[64]。乔姆斯基的形式语法理论以及摩尔定律使得通过语料库来进行统计学习的自然语言理解技术成为可能[65]。近些年来，越来越多的相关研究投入到无监督学习或者半监督学习方法上，这些技术能使自然语言理解不那么依靠需要大量人力获得的人工标注的语料库。一般来说，与有监督学习相比，无监督学习或者半监督学习的难度相当大[65]。

在目前主流的对话系统中，自然语言理解模块通常通过以下流程逐步完成：词法分析、句法分析、语义关系抽取、语义理解。词法分析主要是将字符序列转换为标记序列的过程，并非本文的研究内容，因此这里从其他几方面来介绍各流程的研究现状。

1. 句法分析

通常自然语言理解是通过对句子的某些，或全部句法法进行分析来完成的。大量的形式化体系和算法通过特定的语法体系对句子的句法结构进行解析。大体来说，主流的句法分析包括“依存句法分析”[66,67,68]以及“短语结构句法分析”[69,70]。短语结构句法分析依据短语结构文法[71,72,73,74]，首先对句子进行短语分析，然后指出单词和短语之间的关系，以及短语之间的关系；而依存句法分析依据依存文法[75?]，仅仅在句子中的单词之间标注（有标记的）连接关系。依存句法分析是我们要在本文中探讨的语法类型。这里的“依存”是指语言单位（如单词）由有向链接相互联系在一起。一般来说，在依存语法中，（限定）动词被视为句子或子句结构的中心，其它所有句法单位（单词）是直接或间接地与动词通过有向链接相连。这种有向链接被称为“依存”。句子结构是由一个词（中心词）与它的依存词之间的关系而决定的。“依存”关系是一对一的对应关系：对于句子中的每个元素（例如：单词或语素）来说，实际上句子中都有一个与其相对应的节点。这种一对一的对应关系决定了“依存语法”就是单词（或语素）的语法。所有的句子都有元素和将元素组成结构的依存关系，这种情况应该与短语结构语法的“成分关系”进行比较，“成分关系”是一对一，或一对多的对应关系，也就是说，对于句子中的每个元素来说，有一个或多个与其对应的节点。这种不同带来的结果是：相比短语结构，依存结构非常紧凑，因为它往往包含很多小的节点。从计算机处理的角度来说，这种简洁的结构是有益的。

1. 关系抽取

自然语言理解中的一个重要方向是关系抽取。在NLP领域中，关系抽取已成为一个重要的研究方向，一方面是因为它的实际应用价值，另一方面因为它被看做是语义分析的一部分，而且是目前的技术相对容易实现的那部分。到目前为止，吸引最多注意力的关系抽取是命名实体之间的关系识别，例如：“个人从属关系”和“组织地址关系”。一般来说，关系可以由一个元组的形式来定义的，t=(e1,e2,...,en)。在这里，ei是文本中预定义关系r的实体。大多数关系抽取系统主要关注二元关系的抽取。例如：

位于（厦门，中国）

father-of(Richard Li,Li Ka-Shing)

抽取“高阶关系”也非常有意义。例如这个句子：”At codons 12,the occurence of point mutations from G to T were observed”(“在密码子12，观察到从G到T的点突变”)。句子中出现了一个4进制生物医学关系，可以描述为：

point mutation(codon,12,G,T)

目前人们普遍视“关系抽取”为一个有监督分类问题，它从一个语料库开始（语料库中包含由人类标记的相关语义关系），然后利用统计方法来对标记的关系建模，并学习出适合应用于其它文本的统计模型。目前，关系抽取系统的主要限制在句子层面上运用。事实上，关系可能跨越句子，甚至贯穿不同的文件。然而，解决这一问题需要一定的常识和非常灵活的知识表示。本文的研究也无法直接解决这一问题，但我们相信：通过将句子意义映射为通用的知识表示，我们已经奠定了解决问题的基础。在这种知识表示中，跨句或跨文本的通用联系和推理是可以被实现的。

1. 流结构语法

将流结构语法（FCG）与本文使用的语言形式化体系进行比较分析是很有意义的。在FCG中，一句话语的信息是以语义和句法结构组织在一起的。语义结构是对“话语意思”的分解，它含有特定语言的语义分类，例如：“put”事项被归类为“起因-移动-位置”类事项，包括一个施事者（agent）、一个受事者（patient）和一个位置（location）。句法结构是将话语形式分解为成分和词素，还包含附加的句法分类，比如：句法特征（例如：数量和性别）、词序限制等。

从理论上来说，FCG是基于一种程序性语义的方法，在这个意义上，话语的意思是听话人假定要执行的程序，因此，概念化便是一个规划过程（规划该程序），而解析便是对该程序的执行过程。FCG中有关配对句法和语义结构的例子（短语“the ball”），请见图1.4和1.5。

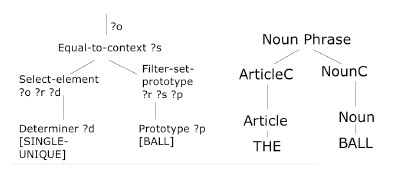


图1.4 FCG中对短语“the ball”的句法及语义表示结构

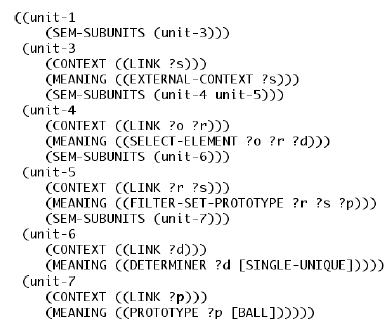


图1.5 FCG中对短语“the ball”的语义表示的列表形式

所有的FCG规则都是双向的。通常在产生过程中，所要表达的语义内容是与语义结构相统一的，有可能产生一组绑定。如果成功了，绑定会与语义结构相融合。这种融合可以理解为“部分统一”，但它利用结构中那些遗漏部分扩展了结构。在句法分析过程中，被分析的句子与句法结构是统一的，同时，结果中的某些部分被添加到语义结构中。

本文所使用的形式化体系在概念上有些类似FCG。此外，我们有配对句法和语义结构，而且还进行双向处理。在第五5章中，我们将论述链语法，它将句子转换成句法结构，以及RelEx和RelEx2Logic模块，它们将句法结构转换成语义结构。在第六6章中，我们将阐述另一个方向的Microplanner和SuReal模块，也就是将语义结构转换成句法结构，再生成句子。本文研究的开源平台OpenCog中的模式匹配器（Pattern Matcher）在使用这些句法和语义结构时，也将这些结构视为有效的程序，同样实现了“程序化语义”。但我们的形式化体系与FCG的着重点不同。FCG主要是用作探索问题的理论工具，而本文的研究更侧重用于真实世界的实际应用。

1. 自然语言生成Natural Language Generation

随着统计机器学习技术不断成熟和发展以及大数据的利用率不断提高，统计的方法已经被广泛用于自然语言理解的很多模块，如句法分析、语义关系抽取，甚至对话系统中的对话管理模块，并取得了很大的进展；然而，由于语义标注语料的缺乏，数据驱动的自然语言生成技术还面临着诸多困难和挑战，目前大部分自然语言生成系统仍然采用了基于规则的方法[76,77]。

自然语言生成（NLG）是自然语言理解（NLU）的反向过程。总的来说，自然语言生成是从信息的计算表示中自动生成人类（自然的）语言。由于语义知识表示体系的差异化，造成自然语言生成系统的输入多样化，因此不同的自然语言生成系统所包含的组块一般都不尽相同。概括地说，自然语言生成系统可以表示为解决下列两项任务：

* + - 我应该说什么？
    - 我应该怎么说？

为了解决这些问题，自然语言生成系统可能涉及许多相互关联的规划模块，如：

* + - 确定要表达的信息
    - 构建语篇规划
    - 将信息块转换为语篇单位
    - 选择适当的短语和单词
    - 输出正确的语法

将这个流程分解为阶段的方法如下：

1. 宏观规划

宏观规划涉及选择和组织内容。它输入的是一个或多个沟通目标：解释、描述，或提问；引起听众的某种行动或思考等。宏观规划输出的是一种知识架构，这个架构不一定是语言的属性，而是体现智能体所要沟通的信息。除了一般性内容，这种知识架构可能包含一些与沟通过程有关的信息，例如：不同的知识块应该以什么样的顺序来传达。

目前宏观规划的研究主要基于一些语篇结构层次上特定的理论体系，例如：修辞结构理论[78]。在本文所介绍的研究中，我们采用了基于动机认知模型Psi的宏观规划，该模型的构建非常符合人类的心理特征。

1. 微观规划

微观规划则运用知识架构，并将它们分为句块。微观规划必须处理多种语言问题，例如：

* + - 句子范围：是否可以将两片叶子接在一起，怎样接在一起。(“我今天饿了。我去了Burger King。”VS“我今天饿了,所以我去了Burger King。”)
    - 代词化
    - 聚合：删除重复内容。（“抽烟对你不好。抽烟会缩短你的寿命。抽烟让你口气不佳”VS“抽烟对你不好、缩短你的寿命，而且让你口气不佳”）
    - 主题化
    - 主题排序

到目前为止，大多数微观规划都是为特定的应用而专门制定的。比较典型的多用途微观规划框架如SPUD[[79]，它利用的是一个基于逻辑的规划流程形式。尽管在细节上有所不同（原来的SPUD是基于树-邻近语法），但我们所描述的微观规划在概念上受到了SPUD的很大影响。

1. 表层生成

最后，“表面生成”涉及的是文本表层的生成，例如：把知识结构转换为句子。经典的表层生成器，例如FUF/SURGE[80]和Penman/KPML[81]，都是以深层语言学理论为基础的。其他知名的系统，例如Nitrogen[82]则采用的是统计法。所有基于规则的方法和统计法现在仍然流行。

在对话系统方面的应用方面，自然语言生成的主要任务是将对话管理生成的抽象对话行为组块转换成恰当的应景的自然语言文本，通常，这里的抽象对话行为组块包含对话行为类型以及相应的（属性，值）二元组，如图1.61.8所示[83]。有关这方面的研究，早期的方法有使用N-gram语言模型来对一些人工编写的语言生成器产生的可能输出结果进行重排[82]，显然该方法需要依靠大量人力来编写规则来构建语言生成器。为了减少人工编写的语言生成器的使用，[84]使用基于N-gram的语言模型生成器组合来替代了这些人工编写的语言生成器，针对每一个对话类型定义一个相应的N-gram生成器，虽然[84]中将人工编写的规则的使用限制在后处理的小集合上，但还是无法避免生成器产生冗余不必要的短语和句子的情况，从而使计算复杂度相当大，而且N-gram的方法也无法判断输出的结果就是包含对话行为组块中最多相关语义项的句子或文本。[85,86]中提出了基于统计的以短语为生成单位的自然语言生成方法，该方法使用了统计方法在带标注的话语语料库中训练出（话语，粗糙语义概念）的二元组，从而根据抽象对话行为组块的属性值得到相关的短语或句子。该统计方法在一定程度了解决了需要大量人力来编写规则的难题，但是带语义概念标注的语料库并不成熟也很难收集，而且统计方法相对耗时，因此使得该方法很难扩展到新的领域。

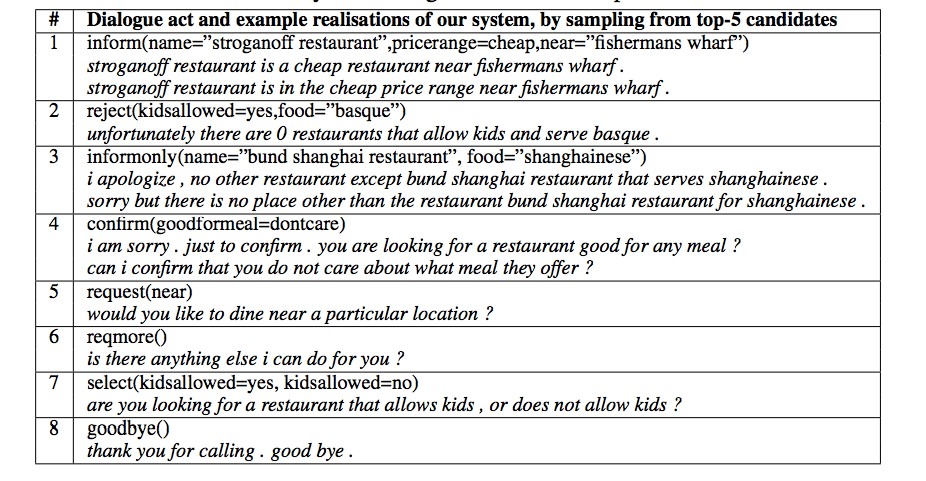


图1.6 对话系统中自然语言生成模块的抽象对话行为实例

针对上述这些问题，[83]将深度学习的方法应用到自然语言生成领域，使用递归神经网络（RNN）和卷积神经网络（CNN）的结合来训练对话行为类型和话语的二元组。该方法无需任何的语义对齐二元组或者预先定义的语法树等标注语料库的指导，因此可以在任何领域的对话语料库上进行模型训练，使其扩展性得到了很大的提高，与上述提到的系统相比，该方法得到的实验结果也有明显的改善。但该方法只是训练出对话行为类型和话语的配对，无法用于针对抽象的语义表示形式的语言生成，然后一个好的认知型对话系统中，要求的对话往往需要通过语义分析和一定的逻辑推理来生成。

总的来说，与自然语言理解相比，自然语言生成技术要落后得多。目前的实用系统往往限定在一些专业领域，而且很粗糙。概念比较先进的系统，如FCG，尚未经过广泛地实践测试。正如第6章中所介绍的，我们在这一领域的研究代表了“基于规则的方法”和“统计法”的融合。我们认为这种融合非常有前景，但目前还没有精细化，也没有经过系统性评估。

### 言语行为理论及相关研究综述

我们认为，分析和管理对话系统需要产生的各类话语的一个有效方法是：使用由Austin[87]提出，Searle[88]改进的言语行为理论（Speech Act Theory）。本文将要阐述的认知对话系统研究中正采用了此方法，且在研究过程中，我们发现这个理论是极其有用且方便使用的。我们在这一节先回顾言语行为理论的基本观点，并介绍一些言语行为理论在对话方面的相关研究以及更细致具体的分类体系。

言语行为理论是语言学和语用学等研究中的一个重要理论。言语行为理论的核心概念是：强调语言的意向功能，认为说话者在说话的同时是在实施某种行为。因此，根据言语行为的观点，分析说话者的语言行为的过程，也就是说话者分析为了实现某个特定目标而采用的一系列的言语行为的过程。

言语行为理论探讨那些可以通过言语来表现的行为类型。Austin[87]将言语行为分为以下几种：

* + - 言内行为（Locutionary Act）：话语的外在表现形式，即：话语本身的字面意义。
    - 言外行为（Illocutionary Act）：话语的“言外之力”，即：说话者对受话者的影响。
    - 言后行为（Perlocutionary Act）：话语达到的实际效果，即：话语所导致的行为，如说服、劝说、吓唬、启发、激励，或者让受话者去做某件事，或者让受话者意识到什么，不论是有意还是无意的。

Searle对Austin提出的言语行为理论做了深入的研究和探讨。在Searle[88]看来，言语行为有时仅仅指的是言外行为。他认为，说话人通过他们的言语只能获得5类言外之力，它们分别是：

* + - 断言类：说话人自己承诺事情是真的。The sky is blue.（天是蓝色的。）
    - 指令类：说话人试图让受话人做某事。Clean your room!（打扫你的房间！）
    - 承诺类：说话人对未来的行为做出承诺。I will do it.(我将会做这件事。)
    - 表达类：说话者表达了某些心理状态。I’m Sorry.(对不起。)。
    - 声明类：说话人带来了不同的状况。The meeting is adjourned.（这个会议休会了。）

受这个本体论的启发，Twitchell和Nunamake在他们2004年的论文（标题为：言语行为归档：分析持续谈话和其参与者的概率统计法。[89]）中提出了更加精细的“42种言语行为”本体论，叫做SWBD-DAMSL(DAMSL=Dialogue Act Markup in Several Layers）。尽管有少量不符合Searle的观点，并被分类为“其它”，但几乎他们所有的42种言语行为类型都能够被映射到Searle的5个高级类别之一。图1.71.9和1.81.10描述了这42种行为和它们与Searle分类之间的关系。我们已经利用了这个Searle解析法，它给我们正在进行的自然语言对话研究带来了灵感。

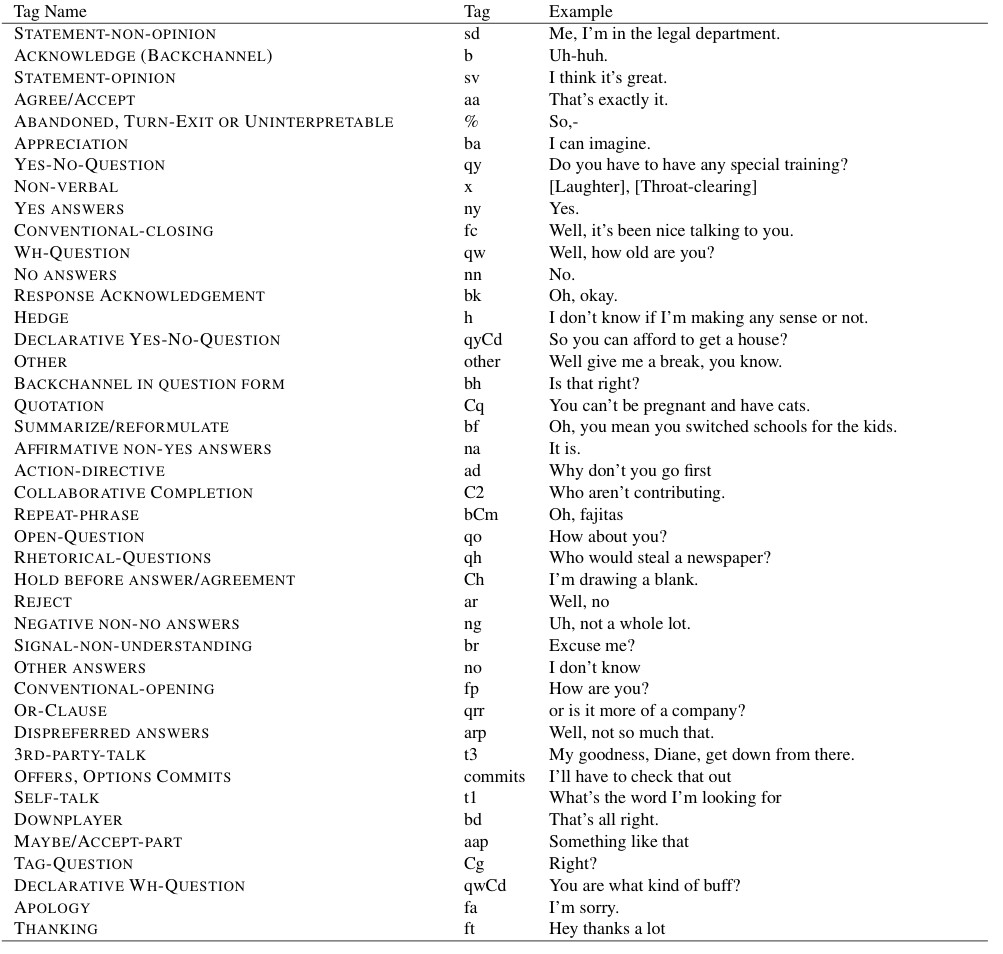


图1.7 SWBD-DAMSL中使用的42种言语行为

总之，言语行为理论为对话管理和对话控制提供了一个新的思路，也就是将对话的话语看成是一种行为，这样不仅能通过对不同言语行为的定义来猜测说话人的意图，还能将对话行为看成和其他非语言的行为类似。不难看出，它不仅给对话管理和对话控制提供更多可行的用于非语言的行为控制机制，也同时能将对话与智能体的其他行为有效结合起来，从而完成更智能更人性化的对话。

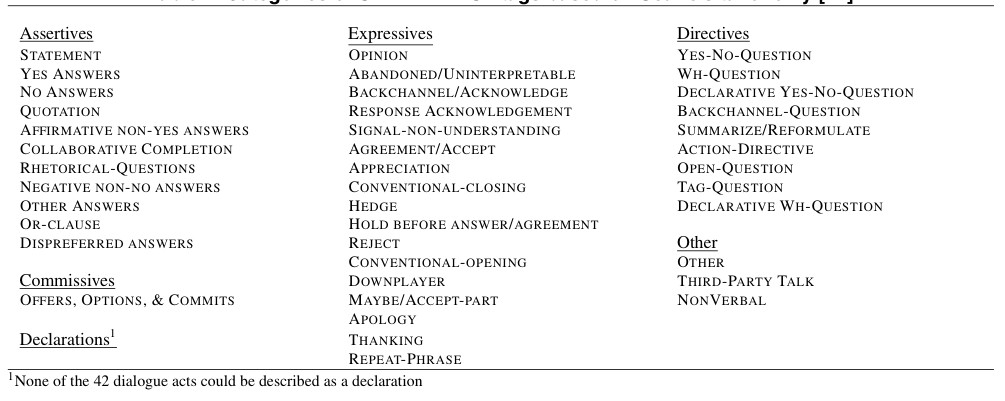


图1.8 42种SWBD-DAMSL言语行为类型与Searle的言语行为类型的关联

## 存在的问题

对话系统的研究已有几十年的历史，尽管有不少学者，这些对话系统能够满足一些娱乐性的需求和非常限定场合下的信息查询用途，但仍有许多亟待改进的地方，主要包括：

**（1）知识深层表示问题**

认知对话系统首先需要一个能存储丰富语义信息且能灵活操作的深层语义表示体系来支撑，这样一来，对话系统就能从自然语言话语中捕获浅层的句法信息之外的深层语义信息，用于指导对话系统的认知过程。 然而知识表示体系的研究面临着表达能力和计算效率两大挑战，目前很多知识表示体系不得不为了使计算效率达到可用范围而牺牲其表达能力，也就是说，只能表达小规模的知识，一旦知识库达到一定规模，往往出现很多不可操控的问题。

**（2）不确定性逻辑推理问题**

不少对话系统中不考虑非精确条件下的语言理解问题，以及在不确定的条件下对话系统如何推断和决策问题。我们认为，具有认知功能的对话过程不仅需要具备一定的逻辑推理功能，即通过演绎、归纳和回溯等推理过程实现的推理，还需要具备不确定推理功能，即涉及概率和模糊逻辑等不确定性的推理，从而使得对话系统能允许非精确输入，并实现在非精确条件下的有效理解，提高对话交互的自然度。

**（3）语言和涉身知识的融合问题**

目前的对话系统基本上只考虑语言层面上的交互，但是，自然的对话交流都需要考虑特定的语境和语用等方面的信息，语言的交互也不仅仅是语言层面上的交互，还需要理解语言背后涉及到的涉身交互，也就是说，每一句话语应该被理解成一个“言语行为”，即包含一些独特的语言属性，还包含一些涉及到言语行为和其他类型行为的语用属性，也就是通常所说“言外之意”。

**（4）动机驱动的对话控制问题**

大多数对话系统都是作为一个执行主体来完成用户指派的对话任务，然而要实现自然的对话交互，还要求机器能作为一个认知主体，能去适应人的交流方式。我们认为，认知对话系统应该根据智能体本身的动机来做决策，并且能在交互的过程中根据人的方式来调整系统动机，以实现更拟人更逼真的对话。也就是说，对话系统在决定什么时候说什么时，必须有一个动机驱动的行为选择模型来指导，而不是基于简单的浅层语言提示来进行回应。

## 研究目标和内容

构建一个能达到人类水平的对话系统是一项十分艰巨的任务，实现这个长远目标远超过本文所能涵盖的范围。本文针对当前相关研究存在的问题，提出了本课题的研究目标：从对话系统的认知需求着手，深入研究知识表示的深层模型以及在该表示上的不确定概率推理，设计并实现自然语言理解和自然语言生成的流程，实现架构从语言到逻辑之间的桥梁。采取动机驱动的情感计算模型和言语行为理论的思想指导，提出一个能使对话系统具有一定认知能力的动机驱动的对话控制模型；并在该模型上实现一个以信息查询为动机的问答系统。

本文的主要研究内容如下：

（1） 研究对话系统的认知需求，借鉴心理学动机模型、言语行为理论以及概率逻辑推理理论等，提出一个动机驱动的对话控制模型。

（2） 研究知识的深层表示问题，使其不但能有效储存丰富的语义语境信息，还能被灵活操作。

（3） 研究基于深层知识表示的自然语言理解机制，设计并实现一个自然语言理解框架，使其能将自然语言语句转换成（2）中使用的基于超图的抽象的逻辑语义表示形式。

（4） 研究基于深层知识表示的自然语言生成机制，设计并实现一个自然语言生成框架，使其能将基于（2）的深层表示的抽象逻辑语义形式转换成自然语言语句。

（5） 研究不确定性逻辑推理，将概率逻辑网络应用在深层知识表示结构中，设计和实现一个能在自然语言语句上执行的常识推理系统。

（6） 借助上述（3）中的自然语言理解框架，将简单的英文维基百科进行自然语言理解处理并表示成基于超图的深层语义表示形式，并在（1）中的对话控制模型上实现一个以信息查询为动机的问答系统。

## 论文的组织结构

本文按照如下方式组织：

第一章，给出了认知对话系统的研究背景和意义，总结了国内外相关领域的研究现状，并指出了现有研究的局限和存在的问题。最后，给出了本文的研究目标，并介绍了本文的主要研究内容。

第二章，深入探讨了认知对话系统中所使用的深层知识表示体系，并从范畴论角度出发，阐述了能将自然语言映射到该表示体系的理论基础，最后，我们还介绍了能用在该表示体系上不确定性逻辑推理机制，即概率逻辑网。

第三章，主要介绍了动机驱动的情感计算模型Psi，以及该模型中所采用的情感和人性等机制，为下一章节的动机驱动的对话控制模型提供一定的背景知识和理论基础。

第四章，本章首先讨论了如何对心理学上动机驱动的情感计算模型Psi进行改进，使其能用在本文的认知对话控制机制中，然后介绍了认知对话系统的相关目标和动机，提出了引入逻辑推理的动机驱动的行为控制策略，并提出了使用动机驱动来进行对话语篇管理；结合言语行为理论，针对不同的言语行为类型提出了不同的言语行为规划器；最后，结合上述研究成果，提出了动机驱动的对话控制模型。

第五章，介绍本文设计实现的将语言转换成超图表示的抽象逻辑形式的自然语言理解流程框架，借助链语法输出的链集合，将其映射到使用超图来表示的更抽象的逻辑关系集合。

第六章，介绍本文设计实现的将超图表示的抽象逻辑形式转换自然语言语句的自然语言生成流程框架，通过微观规划器来重新组织和整理超图表示的抽象逻辑关系，并利用基于超图匹配的表层生成器将这些整理后的抽象逻辑关系转换成自然语言语句。

第七章，介绍本文设计的对自然语言语句进行逻辑推理的机制，结合本文设计和实现的自然语言理解框架，将自然语言语句转换成超图形式表示的抽象逻辑关系集合，并使用概率逻辑网络进行推理，最终将推理后得到的超图表示的逻辑关系集合转换成自然语言语句的形式，特别给出了对带比较级的句子进行推理的过程。

第八章，结合第五、六、七章的研究成果，将简单的英语维基百科上的知识表示成第二章中的超图形式，并在第四章的动机驱动的对话控制模型上，设计并实现了一个以信息查询为动机的基于概率逻辑推理的问答系统。

第九章，对全篇论文的研究进行总结，列出本文的主要贡献和创新点，并提出下一步的研究方向和对未来研究的展望。

第一章 绪论 1.5 论文的组织结构

第二章 智能会话系统中的知识表示和逻辑推理

要使计算机能理解并灵活运用人类的自然语言，将自然语言转换成一种能容易通过计算机来操作的知识表示形式是不可忽视的，这种知识表示体系不仅要满足自然语言的复杂特性，使得自然语言处理系统能在自然语言和知识表示之间相互转换，以实现人机之间的自然语言通信。同时，为了构建一个能较好的与人类沟通的智能对话系统，这种知识表示机制还需要能被应用在适当的逻辑推理系统中使其能进行必要的逻辑推理，以生成能被人类理解的智能的对话应答。

本章提出了一种能满足上述要求的基于有向加权超图的知识表示体系，阐述了该知识表示体系中的基本操作原理和基本数据类型，并从范畴论角度分析了其被应用在智能对话系统中的理论基础。这样的知识表示体系不仅为自然语言过度到逻辑语义形式提供了平台，而且方便了常识推理和自适应学习机制等研究工作的开展。另外，本章还阐述了能被应用在这种基于超图的知识表示体系上和本文研究的智能对话中使用的逻辑推理系统—-概率逻辑网络。

值得一提的是，这里介绍的知识表示体系并不仅仅是适用于自然语言处理和智能对话系统，该知识表示体系目的在于模拟人脑的知识表示方式、记忆空间以及认知过程等等，因此同样适用于其他智能处理模块。本文的研究目标之一也是通过探索智能对话系统来研究实现一个相对智能的人机交互系统。

## 基于超图的知识表示方法

对于任何以实现复杂功能为目标的自然语言处理系统（如智能对话系统或复杂的问答系统等）来说，系统内部知识如何表示是至关重要的。高效的知识表示机制不仅能使系统不同模块之间高效地传递有用的信息，更能使系统的知识库通过与用户的交互不断完善。本节主要阐述一种高效灵活的基于超图的知识表示方法，正如上面提到的，这里介绍的知识表示体系旨在模拟人脑的知识表示方式，因此会先讨论其记忆空间以及全局局部知识存储等问题，然后再讨论超图的数据结构以及基于超图的知识表示体系，同时为了方便后面章节对本文具体的研究工作的解释，在这一节我们还会列出一些该知识表示体系中基本的且后文经常会用到的数据类型。

### 知识表示和记忆空间

1. 本地和全局知识表示

本文所采用的知识表示机制其本质是一个超图表示的网络。把心智当作网络的观点其实是隐含在模式主义哲学中的，因为每一个模式都可以被看成某物的模式，或关于某物的布置的模式——因而一个模式总是可以被看做二个或更多物体之间的关系。一系列的模式形成一个模式网络。各类知识可以以网络形式来表示，而认知过程也可以被表示为网络，举例来说将它们表示为程序，以各种树或图的形式来表示。在一个智能中模式的涌现可以被看作自身的一个模式网络；在涉身的心灵和它的物理和社会环境之间的关系可以被看做某种生态和社会网络。

知识表示体系的两个主要超类是局部（也被称为显示）和全局（也被称为隐式）系统，我们用一个称为全局-局部的混合类包含了这两者。在一个局部系统中，每一条知识都是用一小部分认知系统的元素来存储的；在一个全局系统中，每一条知识都被以一种特定的模式存储与激活，比如以认知系统的一定比例的元素的形式；在一个全局-局部混合系统中，这两种方式被共同使用。以上三种知识表征类型都可以被网络所实现。在本文研究的智能对话系统中，这三者都是以同样的网络（Atomspace）来实现。

1. 记忆类型和相关认知处理

我们的知识表示体系中包含多种的记忆类型，如同上面所讨论的，它的前提是正确地建立一个类人的人工智能系统，以处理不同类型的记忆，这些记忆包括了结构化的和动态的。

该知识表示中的记忆类型有：陈述性记忆、过程性记忆、感知记忆、以及场景记忆，这些在认知科学中被广泛讨论[90]的记忆类型，以及分配泛用的系统资源的注意力记忆、和为特定目标分配系统资源的意向性记忆。这些记忆类型在一个开源的认知体系结构系统OpenCog1[[1]](#footnote-2)中被一一实现，表格2.12.1概述了这些记忆类型，给出了关键的引用并指出了相关的认知过程，同时指出了每一个认知过程（模式创造、联系等）所对应的泛化的模式主义认知动力学。

表 2.1 OpenCog

|  |  |  |
| --- | --- | --- |
| 记忆类型 | 特定的认知过程 | 泛化的认知功能 |
| 陈述性 | 概率逻辑网络PLN2[92];概念调整[93] | 创建新的模式 |
| 过程性 | MOSES（一个创新的概率演化运算学习算法）[94] | 创建新的模式 |
| 场景性 | 内部模拟引擎[95] | 关联，创建新的模式 |
| 注意性 | 注意力经济分配网络（ECAN）[96] | 关联，评分 |
| 意向性 | 概率目标层次按照OpenPsi动机框架通过PLN和ECAN来细化[97]) | 评分，创建新的模式 |
| 感知 | 视觉图像处理组件DeSTIN | 关联，注意力分配，创建新的模式，评分 |

以模式主义认知理论的形式，上述多种记忆类型可被看做特定类型模式的特化存储方式，并经过了计算时间与空间上的优化。联系到某种特定类型的记忆的认知过程被用来创造和识别该记忆的类型。原则上所有类型的模式都可以在统一的记忆和处理构架下进行，这种类型特例化，是为了能够在现有的计算条件可接受的前提下创造有效的泛化智能。我们在[91]中所详细论述过，高效性并不是可有可无的，而是对真实世界中的泛化智能举足轻重的特性（就像Hutter所展示的，如果没有效率的限制，任意登记的泛化智能都可以通过简单而琐碎的程序来实现）。

这种知识表示体系的设计本质在于，与每种类型的记忆相关的数据结构和处理过程是被紧密联系在一起的，相比于仅仅包含同一种结构和处理过程的而仅仅以不同的黑盒分割开的构架，它产生了协同性的智能。我们的知识表示体系中还设计有交互的认知处理过程，以使得不同类型的记忆间可以互相转换，尽管有时这会消费较多的计算资源（比如，一段陈述性的知识可能通过一些努力被解释为过程性或场景性的知识）；同时，对于一个主要处理某一种类型记忆的学习进程来说，它可能会经常通过把知识转换成其他类型来解决问题，比如认知协同任务。

### 基于超图的知识表示库 AtomSpace

正如上述所言，Atomspace是本文所使用统一且通用的知识库，它使用包含特定类型的节点（Node）和链接（Link）的加权有向超图结构来表示知识，主要用作表示叙述性的知识，同时亦间接地表示其它类型的知识。Atomspace中使用的超图的节点和链接类型是经过包括语义层面上精挑细选得来的，以满足智能对话系统在认知过程的需要。由于Atomspace是本文研究中的知识表示核心，在这里我们会详细列出几个简单的示例予以说明。

以下是一个以Atomspace中常用的表示方法去表示继承链接（InheritanceLink）中的例子：

InheritanceLink Lian\_Ruiting animal <.99>

或更明确地：

InheritanceLink <.99>

ConceptNode "Lian\_Ruiting"

ConceptNode "animal "

以及：

EvaluationLink <.7>

chase

ListLink

Cat

mouse

或更明确地：

EvaluationLink <.7>

PredicateNode "chase"

ListLink

ConceptNode "cat"

ConceptNode "mouse"

正如上述示例所示，节点通常具有名称，而链接则没有，但链接可以连接一个或多个的目标，而这些目标可以是节点或链接。

综合来说，Atomspace是一个“带标记的通用超图“，这些标记可以是节点的名称或其真值等等。“超图“与一般“图“的其中一个主要不同之处在于其链接，例如ListLink或SetLink，它是可以连结两个以上的参数，而其通用性也允许这些链接与其他的链接相连，而不只局限于节点。

另一个值得注意的地方是这些节点的名称，虽然在这些示例中它们都是以英语来表示，但实际上我们的系统会在操作中通过自主学习去创造一些新的节点，而这些节点往往都与语言种类无关。

更重要的一点是，AtomSpace的知识表示形式的主要功能是提供一个灵活的方式去紧凑地表达多种相关形式的知识，并允许它们之间有交互操作。其中的“交互操作“是指，例如，一组的陈述性的知识可以跟另一组的注意力相关或程序性相关的知识相互连结，或一组在一个类别的知识可以跟另一组在其他类别的知识重叠在一起（同一链接同时有著一个陈述性相关的真值和一个注意力相关的重要值）。总而言之，只要有任何表示形式能够提供足够的灵活性来：

* + - 紧凑地表达所有在人类记忆中扮演著主导角色关键类别
    - 灵活地创建特定的副表示形式去表达在上述这些关键类别中的知识，同时又能够被迅速地改动或操纵这些知识
    - 重叠和连结不同种类的知识，包括上述这些特定的副表示形式

以上几项都符合我们系统的整体要求。而这些Atom的表示形式已能满足此计设的总体要求，并从软件的角度来看已证明是可行的。

**基本的Atom种类**

Basic Atom Types

* + - ConceptNode – 表达任何的单元，具体或抽象但又不能以其他特定的节点种类 来表达的概念
    - PredicateNode – 表达一个为 Atom 产生真值的程序（以下会有详细说明）
    - SchemaNode – 表达一个以一个 Atom 产生另一个 Atom 的程序，或产生一些其他效果

和以下的链接：

* + - MemberLink – 连结一个通用种类的 Atom 和一个 ConceptNode，并拥有一个模糊真值
    - InheritanceLink – 连结两个 ConceptNodes，并拥有一个概率真值
    - SimilarityLink – 连结两个 ConceptNodes，并拥有一个概率真值
    - EvaluationLink – 连结一个 PredicateNode 以及其参数
    - ExecutionLink – 连结一个 SchemaNode、其需要的参数以及其产生的输出

– ExecutionOutputLink – 连结一个 SchemaNode 以及其输入，当遇到特定 的认知程序时才会产生输出

**绑定程序的节点**

SchemaNode和PredicateNode有两种形态：已绑定和未绑定。未绑定是指系统尚未知道应该如何评定该方案，或系统将会以一连串的ExecutionOutputLinks来绑定该方案。而已绑定的会有一个特定程序的名称（目前该程序可以是C++，Scheme或Python语言编写的程序）来执行该方案。例如：

ExecutionOutputLink

GroundedSchemaNode "plus.py"

ListLink

NumberNode "2"

NumberNode "3"

当执行时，会调用Python函数“plus“来把“2“和“3“加起来，然后产生一个名为“5“的NumberNode作为其输出。

**內涵与外延的关系**

我们所使用的知识库中的链接类型中亦有多种拥有继承和相似性质的链接，例如IntensionalInheritanceLink和ExtensionalInheritanceLink，我们会在2.3中进一步解释，这里只提一下，以下链接的真值：

ExtensionalInheritanceLink A B

同时亦可被表示为：

SubsetLink A B

是一个条件概率P(B|A)。另一方面，以下链接的真值：

ExtensionalInheritanceLink A B

是一个条件概率P(prop(B)|prop(A))，当中prop(X)表示在PLN推理系统中定义的模糊集合X的属性。

**SatisfyingSet**

SatisfyingSet的算符能够表达一个集合关系的概念，而当中每一位成员都是乎合同一个述语的元素，并用Member的关係形式，以一个模糊数值（而非概率数值）来表达该元素属于一个概念的真确性。举例来说，假设现在有“FriendOfBob”这个述语和三个元素“Jack”、“John”和“Jill”：

Evaluation <.7>

FriendOfBob

Jack

Evaluation <.6>

FriendOfBob

John

Evaluation <.8>

FriendOfBob

Jill

根据 SatisfyingSet 算符的定义，我们会得出以下的 Member：

Member <.7>

Jack

SatisfyingSet

FriendOfBob

Member <.6>

John

SatisfyingSet

FriendOfBob

Member <.8>

Jill

SatisfyingSet

FriendOfBob

**真值Truth Values**

一个 Atom 的真值是用来表示一个 Atom 的真确性，在某程度上是取决于 Atom 的类型。这个数值是以一个 TruthValue 物件的形式跟 Atom 链接在一起。一般来说 我们会在 Atom 的表示式后以<> 及一个数字来显示其真值，例如：

A <.4>

來表示一個名稱为“A”的 Atom 的真值为.4。同样地：

IntensionalInheritanceLink Ben monster <.5,.9>

来表示一个连接著“Ben”和“monster”的 IntensionalInheritance关系中有著一个0.5的强度和0.9 置信度。总而言之，<tv> 表示著一个平均值为 tv 的概率分布。这些概念和语义在[[98](#_bookmark217)]中有更详细的介绍。

**量词Quantifiers**

在不确定逻辑理论中，量化过程是一个比较细微的事项。PLN 包含了标準的 ForAll 和 ThereExists 的量词（使用高阶概率定义的概率真值），在处理量化过程时 主要会用 AverageQuantiﬁer 来建构，例如：

AverageQuantifier $X

F($X)

的真值为F($X)的真值的加权平均值，即是以下的总和：

w($X) F($X) / N

**和自然语言处理相关的 Atom 类型Language-Relevant Atom Types**

接下来将会描述一些用来表达具体语言概念的 Atom 类型。 原则上，要处理从 ASCII 编译的语言资料，除了上述介绍的数据结构外就只有以下两个节点类型和一个链接类型：

* + - CharacterNode
    - CharacterInstanceNode
    - ListLink

因而字串就可以以列表或拼接的方式来表达，例如“pig”可以以列表 (#p, #i, #g)来表达。然而，实际上定义特定的語言 Atom 类型也很有帮助的，不仅提高了易读性还在知识推理过程中起着重要的作用。例如：

* + - MorphemeNode/MorphemeInstanceNode
    - WordNode/WordInstanceNode
    - PhraseNode/PhraseInstanceNode
    - SentenceNode/ SentenceInstanceNode
    - UtteranceNode/ UtteranceInstanceNode

## 语言和超图之间的转换之范畴论观

我们的研究目标一直是探讨有关的语言现象和计算系统之间的交集，从而构建具有实用价值的自然语言处理系统，如智能对话系统。后面章节会通过介绍具体实现算法及相关应用来从语用计算语言学角度阐述本文的研究的可行性。本节主要在广泛的数学背景下，语言的范畴论来阐述其理论可行性。

### 范畴论的基础知识

范畴论[99]常被用于形式化各种数学结构中的共同特性，将这些数学结构的概念形式化成一组组对象和箭头（也称为态射）。一个范畴包括两个基本属性：对象之间的箭头可以复合；每个对象有一个标识箭头指向自己。对象和箭头可以是抽象的任何类型，这个简单的结构安排有着非凡的数学能力，使其能被用在探索各个领域的数学理论基础。

形式上，一个范畴包含一下内容：

* + - 一个对象类ob(C)，其中的元素称为C中的对象。
    - 一个箭头类hom(C)，其中的元素称为C中的箭头或态射。hom(a,b)中的元素f称为从a到b的态射，即*f*：*a*→*b*
    - 一个用于操作态射复合的二元运算子◦,使得对于ob(C)中的任意三个对象a，b和c，*f*：*a*→*b*和*g*：*b*→*c*的复合可记为*g*◦*f*或*g f*，并且使得以下两条公理成立：
* 结合律：如果*f*：*a*→*b*,*g*：*b*→*c*和*h*：*c*→*d*，那么*h*◦(*g*◦*f*)=(*h*◦*g*)◦*f*
* 恒等性：对每一个对象x，存在一个恒等态射(identity morphism)1*x*：*x*→*x*，使得任意态射*f*：*a*→*b*，有1*b*◦*f*=*f*=*f*◦1*a*

任何一个有向图都能生成一个小范畴：其中对象为有向图的节点，态射为有向图（视需要也可扩展为有向循环图）中的路径，态射的复合则为路径的串联。这样的范畴被称为由图产生出的“自由范畴”。

### 语言与超图之间的转换之范畴论观

在简单介绍范畴论的基本知识后，这一节主要解释范畴论如何能作为一个整体框架，用于形式化和解释本文研究的语言理解、生成以及逻辑推理等算法。这个形式化过程的关键便是使用基于超图的知识表示方式来做为各模块的共同知识表示框架。

首先我们来引入一个三元组（原子，时间间隔，原子状态），且称为“t-Atom”。大部分情况下，原子状态包括真值（TruthValue）和注意度（Attention Value），以及一些恒量，如名称和类型。

那么在一定时间间隔下T，t-Atomspace包含一组t-Atom，即(A,I,S)，其中：

* + - 在某个时间点t，且t∈T，基于超图的知识表示库中存在原子A
    - I是原子A的状态变化之间的时间间隔
    - S是原子A在时间间隔I期间的状态

因此给定一个T-Atomspace，就可以形成一个“活动图”（activity graph），其中图的节点为tAtom的集合，边则记录图中认知过程变化的活动（activity）。从节点N1到节点N2的边表示一个以N1为输入且以N2为输出的认知过程。各个认知过程组成的信息链就构成了活动图的路径。从范畴论的角度来说，tAtom的集合就是范畴中的对象，而表明认知活动过程的活动图路径就是范畴中的态射。

根据上述的理论基础，本文研究的语言和超图的转换过程可描述如下：

* + - 本文研究所使用的链语法分析器Link Parser（将在第5.1节5中具体介绍）将一个句子中的不同词节点之间的邻接关系转换成链语法中链（Links），从而产生一组链的集合。
    - 本文研究的语义关系抽取工作RelEx（将在第5.2节5中具体介绍）将上述的链集合转换成一组语义关系集合。
    - 本文研究的逻辑关系抽取工作RelEx2Logic（将在第5.3节5中具体介绍）将上述的语义关系集合转换成能适用于逻辑系统进行语义推理的更抽象的逻辑关系集合。
    - 本文研究所使用的逻辑推理工具概率逻辑网PLN（将在下一节中具体介绍）将上述的从自然语言中抽取到的知识与超图知识库中的其他种类的知识相关联。
    - 本文研究的基于超图模糊匹配的问答系统（将在第8章8中具体介绍）将一组原子集合映射到另一组完全匹配或者近似匹配的原子集合。
    - 本文研究的微观规划器Microplanner（将在第6.1节6中具体介绍）将一组系统想要通过自然语言清晰应答的原子集合S分割成需要立即应答的原子集合S1以及可能在不久的将来要应答的原子集合S2。
    - 本文所采用的表层生成器SuReal（将在第6.2节6中具体介绍）将一组逻辑语义关系的原子集合转换成一组链语法关系集合即链集合，然后将链集合转换成词节点之间的邻接关系，从而生成相应的句子。

因此，广义来讲，自然语言理解、推理/匹配，以及自然语言生成的过程可以看成是可以链接成一个语言应答过程的三个态射，而这个语言应答过程则又可以看成是范畴论中的单一态射。

语言应答的质量可以通过对活动图的每一条边分配一个“成本”来量化，其中“成本”与“置信度”成反比。如果自然语言理解过程生成的结果具有较低“置信度”，那么这就会导致在自然语言生成或者语言推理过程中付出更高的“成本”。因此，从这角度来看，对话系统的目标可被视为提供一个能达到指定语用目的的最小成本应答。

在某种意义上说，从数学角度来高度概括和分析一个复杂的语言和超图之间相互转化的自然语言处理系统，实用价值有限，毕竟这样的分析并没有提供系统所需的具体转换算法也没有分析特定语言的细节和语境等。但是，这样的分析给我们提供了一个更广阔的视野，使我们能抛开语言的具体细节而站在一个更高的角度来看待自然语言处理的过程，这显然有助于我们更好地提高自然语言处理系统。通过本小节从范畴论的角度对整个自然语言处理过程的分析，我们不难发出，本文的自然语言理解、生成以及推理过程正是一系列按照适当顺序进行超图的转换过程，而这些转换过程也在智能体的认知活动中起到了重大作用。

## 概率逻辑网

将语言转换成逻辑形式的一个主要目的就是可以进行相应的逻辑推理，本文所采用的逻辑系统是概率逻辑网络（Probabilistic Logic Network，以下简称“PLN”）。PLN是一个独特的推理系统，它嵌入了预测逻辑和传统逻辑的组合。[92,100]中对PLN进行了全面具体的阐述及论证。在此我们仅简单介绍用于语言逻辑推理以及我们的智能对话系统所涉及的基本相关知识。

PLN是一个数学和软件框架，被运用在本文研究的智能对话系统中，用于不确定推理、使概率真值与泛化逻辑推理规则的整合成为可能。具体来说，PLN包含：

* + - 一系列推理规则（例如，演绎、贝叶斯规则、变量归一化、演绎推理，等等），每一项都有一个或多个逻辑关系或词项（以Atomspace的原子Atom来表示）作为输入，并计算输出；
    - 特定的数学方程，基于合适的背景前提假设的概率值，计算结论的概率值。

PLN还涉及一条特殊的功能，评估概率值的置信度（证据的分量，或二阶的不确定性）。最后，PLN在软件中的实现需要重要的抉择，要求考虑推理规则的结构化表征、“推理控制”——这种策略要求，在每个特定的实际情况中，判断以何种顺序做何种推理。

### PLN的一个简单概述

PLN的关键因素是它的规则和方程式。总的来说，一个PLN规则有：

输入：一个原子集合（视规则而定，它必须满足某些要求）

输出：一个原子集合

实际上，几乎在所有情况下，输出都是一个单一的原子，而输入则是一个单一的原子或者是一对原子。

特定的原形例子是演绎规则，它的输入是：

X\_Link A B

X\_Link B C

输出如下：

X\_Link A C

在这里，X\_Link可以是继承性链接（InheritanceLink）、子集链接（Subset Link）、隐含链接（ImplicationLink）或延伸隐含链接（ExtensionalImplicationLink)。

一个PLN方程与一条PLN规则同时存在，它表示了输出的不确定性真值，基于输入的不确定真值。例如，如果我们有：

X\_Link A B <sAB>

X\_Link B C <sBC>

那么标准的 PLN 演绎规则告诉我们：

X\_Link A C <sAC>

其中：

sAC = sAB sBC + (1 − sAB ) (sC − sB sBC ) 1 − s

式中，sA表示节点A的真值强度。

在这个例子中，每一个原子的不确定真值通过一个“强度”数值来给出。总的来说，PLN中的不确定真值可以有多种形式，比如：

* + - 单一的强度值，比如0.8，这表示概率或模糊真值，取决于具体的原子类型
    - （强度，置信度）对，比如（0.8，0.4）
    - （强度，数量）对，如（0.8，15）
    - 不确定的概率值，如（0.6，0.9，0.95），这表示概率间的相互评分

### 前向和后向链接推理

PLN中典型的模式的使用是前向链和后向链推理。

前向链表示：

1. 给出一个感兴趣的原子池（列表）；
2. 应用PLN规则到这些原子上，以产生新的原子，最好也是感兴趣的；
3. 将这些新的原子加如到池中，返回步骤1。

例子：“人是动物”和“动物会吸”是两个池中的原子。它们被演绎规则所组合，形成了结论“人会呼吸”。

后向链分为两种情况，第一种：

* + - “真值查询”，给出一个原子目标，它的真值未知（或者过于不确定），以及一个原子池，按照演绎规则，通过组合池中原子，找到一种方法来评估该目标原子的真值。

例如：目标是“人是否会呼吸？”（InheritanceLink people breathe）。目标的真值通过“人是动物，动物会呼吸，因此人会呼吸”的推理来评估。

第二种：

* + - “填空查询”，给定一个目标链接（原子可以是节点或链接）以及一个或多个目标中间的变量原子，找出什么原子可以被放在变量原子的位置上，可以使目标链接获得一个高的置信度（即一个“高的真值”）。

例如：目标是“什么会呼吸”，即“继承链接$X呼吸”直接在原子空间中查找发现院子“继承链接动物会呼吸”，表示空格$X的位置上可以被填入“动物”。推理揭示“继承链接人会呼吸”，因此空格$X也可以被填入“人”。

又如：目标是“什么会呼吸和加法”，即“(InheritanceLink$X breathe) AND (InheritanceLink$X add)”。推理揭示此处$X可以被填入“人”但不能是“猫”或“电脑”。

常识推理可能涉及一个前向和后向链的组合。推理中最困难的部分是“推理控制”——在可能的推理步骤中选取哪些步骤，以获得需求的信息（在后向链接推理中）或获得感兴趣的新信息（在前向链接推理中）。在一个有大量原子的原子空间中有许多可能的和强大的启发信息需要进行选择。推理控制的最佳指导是某些基于系统的过去推理历史的归纳。当然，一个较信的系统不会有很多的历史信息。依靠非直接的相关历史是一个推理问题——这个问题的最好解决是让系统有一些历史经验。

### 一阶概率逻辑网络

我们将在这一小节中更具体介绍概率逻辑网络PLN。PLN被分为一阶和高阶子理论，分别为一阶概率逻辑网（First Order Probablistic Logic Network,简称为FOPLN）和高阶概率逻辑网（High Order Probablistic Logic Network,简称为HOPLN）。这些名称来源于NARS[24]。我们首先使用了FOPLN，然后他们使用了HOPLN。

FOPLN是一个传统逻辑，设计到词项和词项间的关系（链接）。它是一个不确定逻辑，词项和关系都拥有真值对象，真值对象有多种可能的类型，从单一的数值到复合的结构如不确定概率。词项可以是基本的观察，或一个符号集合T中的抽象的符号。

**核心FOPLN关系**

“核心FOPLN”涉及集合中的关系：否定、继承、概率合取和析取、成员和模糊合取和析取。基本观察只能有成员链接，而标志词项可以有任何类型的链接。PLN通过链接不同类型的语义来清晰地区分概率关系和模糊集合关系。成员语义通常是模糊关系（尽管它们可能是脆弱的（crisp？），而继承关系是概率性的，并且有规则来管理这二者的互操作。

假定一个虚拟的主体对一个命叫Fluﬀy的生物做了一次基本的视角观察o。这个主体可能将o以0.9的隶属度划于“毛茸茸的”模糊集合下，也可能以0.8的隶属度将之划于“动物”的模糊集合下，于是该主体可以在记忆中建立以下链接：

Member *o* furry < 0.9 >

Member *o* animals < 0.8 >

随后，该主体可能想要通过合并这些链接来完善它的知识。使用最小化的模糊合取操作，该主体可能会总结出：

fuzzyAND < 0.8 >

Member o furry

Member o animals

这表示对o的观察结果是一个“毛茸茸的”“动物”对象。

（延伸）继承关系的语义与成员关系完全不同，尽管它们是相关的。延伸继承（ExtensionalInheritance）表征一个纯粹的条件概率子集关系，通过子集关系来表达。如果A是“毛茸茸的”而B属于“猫”集合，那么以下陈述：

Subset < 0.9 >

A

B

意思是：P(x 属于集合 B|x 属于集合 A) = 0.9

### PLN 真值

为了增加全概率析取的信息量，PLN拥有一系列不同的真值类型：

* + - 强度真值，包含单一数值；例如，< s > 或 < 0.8 >。强度真值通常表示概率，但不总是这样。
    - 单一真值，包括一对数字。这些数字对以以下形式存在：< s, w >，s 是一个强度值而w是一个“证据的权重值”；< s, N >，其中N是一个“计数”。“证据的权重值”是对信念的量化描述，而“计数”是对重复性证据的量化描述。
    - 不确定性真值，它用区间 [L, U] 、评分水平b、以及一个整数k来量化对真值的描述。不确定性真值量化了以下的想法：在经过了k次观察以后，以概率b的可能性，推理的结论会落在区间 [L, U] 中。
    - 分布真值，对整个概率分布的离散化近似。

**附加的FOPLN关系**

在FOPLN核心关系之上，FOPLN还有两类额外的关系类型。有一类简单的类型，相似度，定义如下：

S imilarity A B

如果R A B的真值可以仅仅用核心FOPLN关系中A和B的关系来计算的话，我们把关系R叫做简单的。而有一类复杂的“附加”关系，如意向继承，描述了与某一词项相关联的模式的属性集合与相应的其他属性集合之间的关系。

回到我们的例子上来，主体可能观察到“猫”的两项属性，即“毛皮”和“会叫”。由于希腊神话中的三头狗Fluﬀy也是皮毛动物，主体可能会认为：

IntensionalInheritance <0.5>

Fluffy

cat

意思是Fluffy拥有50%的猫的属性。为了更深入地构建这种关系，PLN还有一个混合的意向/延伸继承关系，简单地通过如子集和意向继承关系的析取来定义。

如例子所陈述，对一个复杂的附加关系R，R A B的真值是通过一个数值所表达的FOPLN中不同词项的关系的真值来定义的（而不是“A而且B”），它通过某个特定的数学方程来计算。

### PLN 规则和方程

PLN 中一个析取是通过规则和方程来达到的。PLN 逻辑推理以“三段论演绎 规则”的形式进行，以通过合并陈述和匹配的词项来寻找模式。PLN 的规则包括但 不限于以下例子：

* + - 演绎推理 ((A → B) ∧ (B → C) ⇒ (A → C))
    - 归纳推理 ((A → B) ∧ (A → C) ⇒ (B → C))
    - 溯因推理 ((A → C) ∧ (B → C) ⇒ (A → C))
    - 调整，即合并具有不同真值的同一逻辑关系的两个版本
    - 反演推理 ((A → B) ⇒ (B → A))

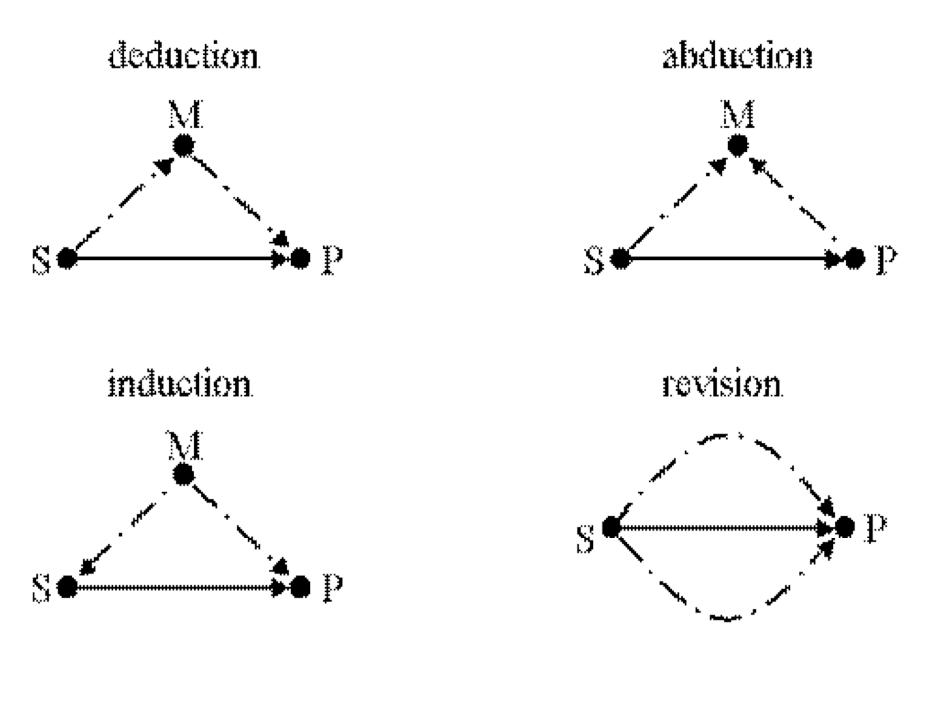


图 2.1 最基本的四个一阶PLN推理规则

这些规则的前四项的基本设计如图[2.1](#_bookmark39)所示。我们可以看到前三项规则表示了 在三个相关联的词项上做推理的规则，同时还可以看到，归纳和回溯可以从前向和 后向链接推理的组合中获得，这是 PLN 真值公式使用的一项规则。

每一项规则都与一个公式相关，它应用规则以计算出真值。例如，假设 sA，sB，sC，sAB，以及 sBC 分别表示词项 A、B、C 以及关系 A → B 和 B → C 的真值，那么， 在合适的条件下，演绎规则的公式如下：

*sAC* = *sAB sBC* + (1 − *sAB* ) (*sC* − *sB sBC* ) , 1 − *sB*

其中sAC表示关系A→C的真值，这个公式的前提是，假设A→B和B→C是相互独立的。

对于仅仅与模糊操作相关的推理，PLN的缺省版本使用带最小值/最大值公式的标准模糊逻辑（尽管还可能有与整体的PLN框架保持一致的变化）。最后，符合并模糊和概率操作的语义在[92]中有按时，但在[101]中有更严格的论证，给出了精确的的语义以构建以下形式；

*Inheritance A B*

其中A和B由前述的成员关系Member C A,Member D B等给出。显而易见，在一个清晰的情况下，所有的成员链接和继承链接的强度都是0或1，FOPLN就退化成标准的谓词逻辑。当继承是清晰的但成员关系不是的时候，FOPLN退化成高阶模糊逻辑（包括词项的模糊表述、以及模糊表述本身的模糊表述，等等）。

## 本章小结

本章主要阐述新颖的并用于智能对话系统中的基于超图的知识表示方法及其灵活性和高效性；以及用于智能对话中的逻辑推理系统概率逻辑网PLN；并从范畴论角度高度概括分析了本文研究的大致框架。本文根据本章阐述的相关理论知识设计并实现了一个基于超图的语言逻辑推理系统，由于该系统依赖与未来章节要介绍自然语言理解系统，因此我们将会在第7章中具体介绍并给出相应的推理实例。

使用这样基于超图的知识表示体系对于自然语言处理和智能对话系统的优点是不言而喻的，它不仅为自然语言过度到逻辑语义形式提供了平台，而且方便了常识推理和自适应学习机制等研究工作的开展。它能通过超图之间的关联抽取了有效的词、句子和语篇等信息，使得自然语言处理能在一个更大的上下文语境中进行，从而使计算机能更有效更智能地处理自然语言；由于该知识表示体系不仅仅适用于自然语言，还同样适用于表示其他感知信息，如可以储存视觉处理后的信息，这就很直观地将一些非自然语言信息加入到自然语言处理系统中，来解决一些通过一般自然语言处理系统无法解决的难题，或者一般自然语言处理系统很难解决的问题。最典型的问题如词义消歧和指代消解问题，一般都是需要根据上下文和相关的语境背景知识来解决，甚至有些需要通过直观的感知信息来解决。比较常用的例子如：

I saw a man with a telescope.

这个句子可以表达“我用望远镜看见一个人。”，也可以表达“我看一个拿着望远镜的人。”。如果计算机想要确定具体是哪一种，可以根据上下文中是否含有“这个人是否拿着望远镜”或者“这个人是否离我很远”等提示，也可以通过图像识别来检验“我”是否拿着望远镜来判定。不管采用哪种方法来消歧，都能体现这样的基于超图的知识表示体系的优势。

除此之外，该基于超图的知识表示体系还适用于2.3中介绍的逻辑推理系统，这不仅能一定程度改进目前的自然语言处理系统，也为自然语言推理提供了一个方便有效平台，这些无疑都是提供计算机智能的有效途径。

第三章 智能对话系统中的情感计算机制

自然的对话过程，除了语言的交流外，还需要情感的交互，情感的交互不但可表现在表情、姿态、手势等方面，还能用于指导对话决策，比如当发现对话者情绪低落时，可以采取委婉的方式继续话题，或者转移话题等。同样，为了实现具有一定认知能力的对话系统，也必须引入情感计算机制，使得对话系统在与人交流过程中表现得更自然更亲近。本文选择了心理学上的动机驱动的情感模型Psi作为理论指导，从对话系统的认知需求出发，对Psi模型进行了改进，使其能被用于认知对话系统中，为对话过程中的对话行为选择做指导。

为了更好地解释本文提出的动机驱动的对话控制模型，本章将简要介绍情感模型Psi的基本理论，包括该模型中的动机行为、情感和个性等机制，以及基于Psi理论实现的智能认知体系结构。

## Psi模型中的动机行为

Psi是由德国心理学家Dietrich Dorner提出的情感认知动机理论模型[102]，将情感与智能体的需求和动机相联系。MicroPsi[97]是一个基于Psi理论的一个开源的智能认知体系结构，实现了Psi理论中的动机、情感以及智能的关联模型，并在一些实用控制应用以及简单虚拟世界里的智能体上进行测试。MicroPsi在全面性以及神经科学和心理学依据方面做得非常出色，但是该认知体系结构在可扩展性上存在不少问题，[103]中认为，MicroPsi目前使用的算法在学习和推理上不大可能被扩展或规模化。

开源的智能认知系统OpenCog（也是本文的认知对话系统研究所用的平台）受Psi理论中的动机和情感模型的启发，借鉴了一些MicroPsi的基本实现方法，实现了类似的情感动机模型OpenPsi。虽然OpenPsi和MicroPsi在一定程度上很相似，但两者还是存在着很大的不同。比如，两者使用了完全不同的知识表达方式，MicroPsi则使用了类似神经元的“quad”来表示知识，每一个quad包括5个神经元，其中一个是核心神经元，其他4个描述与核心神经元的“前/后”或者“部分/整体”等关系。OpenPsi使用了本文第二章中介绍的基于超图的知识表示，显然是一种更灵活和通用的知识表示方式。MicroPsi目前还是注重底层的智能的实现，还未开始着手高层的智能处理，如自然语言处理和抽象逻辑推理。在本节对Psi和MicroPsi的概述中，我们主要介绍其在认知对话系统中会使用到的部分，主要是处理动机、行为和目标的框架模型。

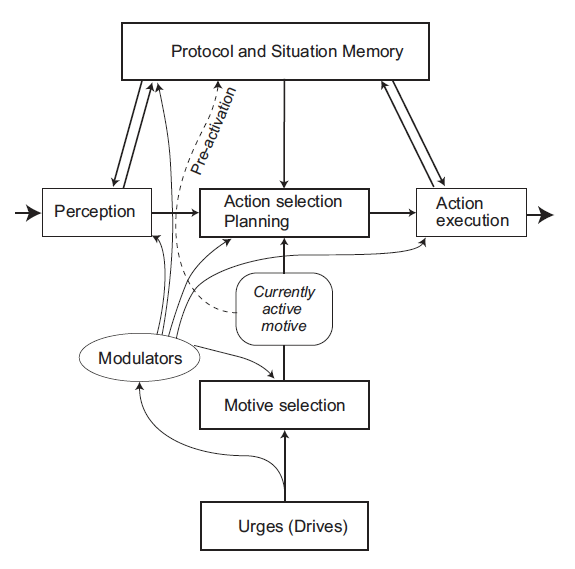


图 3.1 Psi模型的理论框架（摘自 [97])

Psi理论中的动机机制可参考图[3.1](#_bookmark43)，从下往上看，不难发现Psi的动机机制从能激发智能体的需求出发。对于动物来说，该需求可以是食物、水、保护自己的孩子、社交需求、新鲜感等等；对于智能机器人来说，该需求可能是电源、完整性 (保护身体完整, Integrity) 、确定性 (了解和熟悉环境的需求, Certainty) 、认知需求、心里成长等等；对于智能对话系统来说，该需求可以包括收集相关信息、取悦谈话对象、使会话保持新颖不枯燥等。Psi理论特别指出了两种相当抽象的需求，并认为他们是心理学上的最基本需求 (参见图**3.2**）。

* + - 能力需求(Competence)：智能体能有效实现某种强烈欲望的需求
    - 确定性需求（Certainty）：智能体了解和熟悉环境的需求

每种需求都有一定的偏好范围或目标范围，随着时间和环境(包括智能体自身)的不断变化，智能体的需求也在不断地变化。当需求水平落在相应的目标范围时，称作该需求被满足，否则需求不被满足。我们可以将智能体视为一个目标驱动的系统，其主要任务就是尽可能让所有这些需求都被满足。而当某一需求不被满足时，智能体会有一种试图将该需求水平恢复到偏好范围内的愿望，这种愿望便构成了动机(Urge)。

另外还有一种愉悦感（Pleasure）（和其反义的“不悦感”）的原始概念，人们认为这种情绪与复杂的“幸福感”不同。愉悦感被解读成与愿望有关：当愿望（至少部分上）被满足，愉悦感便会由此而生；而当愿望愈趋强烈，不悦感则会由此产生。满足愿望的程度未必要被即刻定义；例如来说，它可被定义为一种需求对近期以来的目标范围随时间衰减的近似加权平均值。

因此，若智能体感到枯燥无趣，当受到大量新鲜感的刺激，它就会体验到某种愉悦感。若智能体感到枯燥无趣，而又更极端地受到单调的刺激，它就会体验到某种不悦感。

要注意的是，根据这相对简单的方法，任何幅度降低了不满感都会造成某种愉悦感；但若一切总是持续在其可接受的范围进行，就不会出现任何愉悦感。这看似有点违反直觉，但必须了解，这些简单的“愉悦感”与“不悦感”并不会完全掌握与这些字词相关的自然语言概念。此处使用的自然语言术语仅作为启发法传达涉及程序的一般特性。这些都是非常低水平的程序，在人类经验的相似情况远低于意识水平。每个需求都有许多参数。如Psi模型所设，其中可调整的重要需求参数有：

* + - 权重：在特定时间点，相对于其他需求，该需求如何被加权
    - 增益：决定源自实现需求所得到的满足感多寡的定标因素
    - 损失：决定源自未实现需求所得到的未足之感多寡的定标因素
    - 衰减：基本上是被给定的愿望随着一段时间的增加率。

在此模型中，为刺激的缺乏使愿望随着时间增加。 本文不确定这个模型是否与一般认知模型一样切实，但针对能确保某智能体持续在运作，这是可行的短期机制。对应多种需求来调整增益、损失和衰竭等参数，便是变换智能体“个性”的一种方式。

接下来，目标被视为系统在未来某时间点致力成真的一种表达；而动机为一组( urge, goal )二元组，由一目标组成，该目标的满足感被预测隐含了某些愿望的满足感。事实上，愿望可被视为顶层目标（在OpenCog中有时又被称作“Ubergoals”），而智能体的其他目标则为子目标。“意图”也被视为综合体：在特定时间点的意图是由积极动机所构成，配合它们的相关目标、行为程序等。

在Psi模型中，智能体随时都有“主导动机”；对我们的认知对话系统而言，虽然也是可行的假设，但大体上是过度限制的假设，比起相似人类或其他高级人工智能系统，此假设或许更适合用在较单纯的模拟动物上。大致上可想成不同动机具有不同权重，而这些权重指出追求上所要耗费的资源量。

Psi模型的基本行动逻辑是通过“三元组”执行，非常类似OpenCog中的三元组Context∧Procedure→Goal。然而，有个重要角色是由四个调制因子扮演，其控制观感、认知和行动选择的程序如何在一特定时间受到规范：

* + - 活化度，决定智能体相对认知、反思活动，侧重于快速、密集活动的程度
    - 分辨程度，决定系统对于尝试解读世界的精准度
    - 确定性，决定系统尝试达到确实、特定知识的努力程度
    - 选择阀值，决定系统对改变其选择侧重的目标的意愿程度

这些调节因子在非常抽象的层次上表现出系统情绪和认知状态的特性；它们本质上并不是情绪，但它们对智能体的情绪有相当大的影响。它们预期的互动如图表3.2所描述。

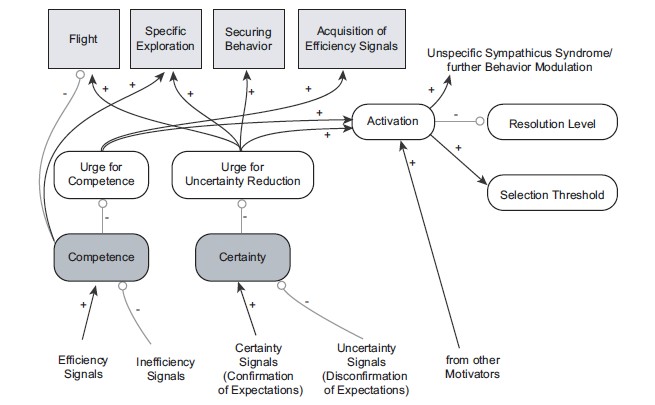


图 3.2 Psi 调节因子之间的主要相互关系(摘自[97])

Psi利用三个网络中安排的Quad储存知识，分别为：

* + - 储存陈述性知识的传感器网络：将图像、物体、事件呈现为分层结构的规划器
    - 运动网络，通过阶层行为程序容纳过程知识
    - 处理需求的动机网络

Psi模型的感知是根据“HyPercept”基于假设的感知的机制，其试图预测待理解的为何，并利用感知和记忆尝试验证这些预测。此外，根据对现实的探索和呈现之间的“Neisser知觉环”，HyPercept密切地联结外在世界的动作。感知上获得的信息会被转换为能够引导行为的规划器，再实施这些规划器（有时会显著影响世界），在进入用来引导进一步感知的程序。我们将不会利用Psi的这个层面，因OpenCog处理感知的方式不大相同。Psi模型的动作选择是根据所谓的“三元组”运行，每个的组成是由：

* + - 侦感器规划器（前置条件、“条件规划器”；如OpenCog中的“语境”）
    - 随后行动规划器（行动、效应器；如OpenCog中的“程序”）
    - 最终侦感器规划器（后置条件、预期；如OpenCog的谓语或目标）

这些三元组可能不完整（三要素之中可能会缺少其中几个）且不确定。然而，在这三元组和OpenCog的概念/程序/目标三元组之间，似乎没有根本上的差异；MicroPsi和OpenCog在此层面的差异，在于用于图解的基本知识呈现，以及用来表达含意的概率逻辑。解出规划器待执行、以达到当前情境所选的目标，会在Psi模型中运用一种名叫“Rasmussen阶梯理论”（以丹麦心理学家Jens Rasmussen之名命名）的程序结合来完成。Rasmussen阶梯理论说明动作的组织，是这三种行为阶段的动作；其中有以技能为基础的行为、以规则为基础的行为和以知识为基础的行为：

* + - 若被给予的任务相当于训练过的例行程序，则会活化自动行为或技能；一般而言可不经由意识注意或慎重控制来执行。
    - 若尚无自动行为，行动步骤可能会出自规则；在可采用一组已知策略之前，必须先对情况做分析，使策略适合用于情况。
    - 若没有适用的已知策略，在这些情况下，必须先找出合并现有操作指令（运算元）到达成所设目标的方法。此阶段通常要求行为的重新构成，也就是规划的程序。用于Psi和MicroPsi执行的规划算法是个相当简单的爬山法规划器。虽有人假设，针对高级智能也许会需要较复杂的规划器；但部分Psi模型的假设，是建立于一旦有机体具有正确的知觉表征和目标，大多数对该有机体进行实际规划所要做的，就相当简单。

在本文的认知对话系统中，采用的认知规划器可对应于Psi模型中的前两种行为（将在下一章进一步说明）。有些感知规划器包含纯反射、自发行为（例如对”你好吗”的问题回答”我很好”）； 而其他的则需要通过大量的推理来封装行为。

## Psi 模型中的情感与个性

情感是人类心理学和人机交互的重要层面。若对话系统能够做到情感的自然互动，对人类谈话对象而言，自动化自然语言对话将更加受人注目且令人满意。要将情感自然度注入到对话中，Psi模型会是合适的方法，因其含有栩栩如生的人类情感模型。Psi中的情感被视为复杂的系统响应模式，而不是清楚构建出的实体。情感是响应某特定愿望所唤起的心理实体。在Bach的论文[]“涌现的情感”（“Emergent Emotions”）中，他将特定的情感与基本的调制器值建立关连，例如：

* + - 愤怒（Anger）：负价、高激发性、低决断水平、选择阈值高
    - 伤心（Sadness）：负价、低激发性、高决断水平、低防御度、低意图性
    - 喜悦（Joy）：正价、高激发性、低决断水平
    - 福佑（Bliss）：正价、低激发性、高决断水平
    - 沮丧（Frustration）：负价、中等激发性、低决断水平、选择阈值低（此为近期才加入到本文的研究中，而前面的项目则是出自前文提及之Bach的论文）

这类关系在概念上的性质或许值得强调。我们并不是在试图将人类的情感以任何意义上“降低”为调制器值的结合。确切的说，我们认为如“愤怒”、“喜悦”等情绪字词，都是对复杂的人类心理动力极原始的叙述。通过适当地合并调节器的值，我们就可经由许多情感字词所示的指出系统动力空间惯常上稍接近的区域。当然人脑/人体的动力系统含有许多未在Psi中建模的复杂参数，我们并不尝试做出精准无误的神经生理情感模型。个性–在对话语境中也是个有趣的话题。若想创建显示多种个性特质的对话系统，可能要以密切相关的方式来处理。经典的人格”五大因素”模型，是使用五种因素来解释人类个性的变异：

* + - 直率性（openness）
    - 尽责性（conscientiousness）
    - 外向性（extraversion）
    - 亲和性（agreeableness）
    - 情绪不稳定性（neuroticism）

“涌现的情绪”（“Emergent Emotions”）文中论证，这些至少可从增益、衰减和损失参数的角度粗略地解释归属感、竞争、确定性和美感等需求。基于这点，可根据多种需求的参数来撰写计算所设人物五大因素每项估计值的程式码。或者也可逆转数学，写出简单的等式，其中含有

* + - 输入：针对五大人格因素的量性度量，与特定人物相联系
    - 输出：针对归属感、竞争、确定性和美感等需求的具体增益、损失和衰减参数值，意图相应于人格五大因素的指定的值

通过对每项人格五大因素（例如个性被指定为五维向量）从范围[0,1]指定一个数字，即可在建构档案中指定一人物的个性。当然与人类个性的错综复杂相比，这显然较粗糙； 但针对适合的特殊应用情况，可以配合提供的对话系统个性内容，在一定程度上这是很实用的工具。

## Psi模型在认知对话系统中的表示

这一小节我们将讨论如何将Psi模型中的动机性行为的基本模块集成到认知对话系统中的。为了使Psi模型能结合对话的上下文信息并参与到知识推理和学习过程中，我们选择将其表示在本文第二章中介绍的基于超图的知识表示库Atomspace中，本文借鉴了 [107]中的一些具体做法。这里只简要罗列出其中的实现要点：

* 需求用GroundedPredicateNodes（GPN）表示，即节点的真值计算由一些内部的C++ 程序或者程序库中Combo 程序来完成
  + 例如： 警觉，外界感知的新鲜感，内在的新鲜感，得到老师的奖励，社交刺激等
* 动机也用GPN表示，其真值根据需求所对应的节点的真值来定义。OpenCog中有时使用Ubergoal 来代替动机（Urge），通常用来表示顶层目标。
* 每个系统都有固定的Ubergoal 集合（且只有非常高级的系统可以修改它们的Ubergoals）
  + 例如： 现在和将来都要活着并保持警惕；现在和将来都要经历和学习新的东西；现在和将来都要从老师那获得奖励；现在和将来都要和其他智能体都要有丰富的社交行为等
  + 一个更高级的系统可以把有抽象（但必须与经验绑定）的伦理原则作为其中的Ubergoal。例如：根据[108]讨论的伦理道德，一个Ubergoal 是追求快乐；一个Ubergoal 是促进成长；一个Ubergoal 是帮助抉择。
* Ubergoal中的ShortTermImportance（STI）表明该目标的紧急程度，因此，如果对应Ubergoal的需求在其目标范围内，那么Ubergoal的STI值为0。所有的Ubergoal都能被给定最高的LongTermImportance（LTI），以保证他们不会被删除。
  + 例如： 如果系统在一个（根据Ubergoal）持续提供足够的新鲜感的环境里，那么对应外界新鲜感需求的Ubergoal会有低得STI值但是高的LTI值，表示该系统不会浪费资源在寻求新鲜感上。但如果环境变得更单调，那么对外界新鲜感的需求的紧急程度就会增加，而其中的STI也会相应增加，资源会开始重新分配在提高新鲜感上。
* 愉悦感也是用GPN表示, 其中的内在真值计算程序将Ubergoal的满足度当成其期望的满足度。
  + 有多种数学函数能用于均衡不同 Ubergoal 的满足度（针对不同p的 p'th power average[[2]](#footnote-3) ），而这里对愉悦感的不同计算方式的选择， 可以带来不同“个性”的系统。目标可以用节点或边来表示。系统的目标列表被称为目标池（Goal Pool）。Ubergoal一般都是自发性的目标，但也有可能有许多其他目标。
* 目标可以用节点或边来表示。系统的目标列表被称为目标池（Goal Pool）。Ubergoal一般都是自发性的目标，但也有可能有许多其他目标。
  + 例如：“从老师那获得奖励”的Ubergoal可以产生类似“从Bob那得到奖励”（如果Bob是老师）的子目标，也可以产生“使老师微笑”，或“创造新的令人惊讶的话语”（如果后者也能得到老师的奖励）的子目标
* Psi中记忆使用本文第二章介绍的知识库Atomspace来表示
  + 例如：认知对话系统在什么语境下说过什么的记忆都会存储在Atomspace中。
* Psi的动机选择通过注意力经济分配机制（Economic Attention Allocation，ECAN[96]）来执行，即对目标节点分配ShortTermImportance
  + 例如：重要值从“从老师那得到奖励”到“从Bob那得到奖励”到“给Bob讲个笑话”的传播流程，是Psi中被称为“动机选择”的一个实例。虽然还没开始执行任何行为，但智能体在这过程中已经对该实现哪个的具体目标做出了选择，而这个具体目标会被用于指导行为的选择。
* Psi的行为选择通过规划器（Schema）来实现，行为选择的问题就可当成选择哪个规划器去执行，更具体地说，在认知对话系统中，选择哪个言语行为规划器（将在下一章中进一步介绍）去执行，而不是选择哪个具体行为去执行。其实，类似的概念在Psi上也存在，Psi中的“自发性行为（automatized behaviors）”类似这里使用的规划器；唯一的区别就是在本文中这些自发性行为是默认的情况。
  + 例如：如果“做一个能惊讶谈话对象的有趣声明”有一个高的STI，那它可被用于激励相关执行程序的选择。如“从以前读过的文本里找到有趣内容，从中抽取摘要，并简明地表达这些摘要”的程序。 当一个程序被选择时，该程序还可能触发相关的子程序的执行。
* Psi的规划通过概率逻辑推理引擎PLN来完成。
  + 例如：当智能体决定说服谈话对象相信某事实X时，它可能需要进行周密的计划，如：推断出谈话对象缺少哪些相关的背景知识去理解X，然后首先给谈话对象传输所缺的相关知识，再来传输X。
* 调节因子用系统参数表示，可以用PredicateNode来表示，且必须在多个动态的智能体之间相互协作。如：
  + 激活度（activation）影响行为选择。当激活度较高时，智能体倾向于做出快速响应外部刺激，因此可能会选择那些能做快速回答的规划器来执行。
  + 解析度（resolution level）影响推理。当解析度较低时，智能体选择减少精力去解析输入中的语言现象。
  + 明确度（certainty）影响推理、模式发掘以及新概念构建的过程。当明确度较低时，智能体接受更多不确定的结论。
  + 选择阀值（selection threshold）引入偏好信息，可以帮助智能体在几个互相冲突的目标之间做出选择。

## 本章小结

本章旨在探讨动机驱动的情感模型的基本原理和工作机制，为下一章节的对话控制模型奠定了坚实的理论基础。 本章首先从心理学上的情感模型Psi出发，分别介绍了其中的动机行为和情感个性等机制，并介绍了基于该模型实现的智能认知系统MicroPsi中的动机行为机制。为了更好地解释我们提出的动机驱动的对话控制模型，本章将本文工作的平台OpenCog中对Psi模型的应用与MicroPsi在动机行为机制实现进行了分析和比较，还简要介绍了Psi模型在认知对话系统中是如何表示的。

第四章 动机驱动的认知对话系统建模

本文研究的长远目标是实现一个具有与人类相当的认知能力的对话系统，而这样的对话系统不仅需要第二章介绍的深层知识表示体系的支持，还需要有一个能处理情感动机以及理解用户潜在意向，并依据这些理解来做出对话决策的认知对话管理体系。因此本文深入研究符合人类自然对话的情感模型和语言语义学理论，旨在构建一个具有认知功能的可灵活扩展的动机驱动的对话管理机制。本章在第三章介绍的Psi模型的理论基础上，结合言语行为理论的指导，并借助概率逻辑网络的强大推理功能，提出了一个动机驱动的对话控制模型。下面将具体介绍该模型的各个模块的理论基础和设计方案，并指出在该对话控制模型上如何实现一个认知对话系统。

## 认知对话系统的概念模型

在介绍本文提出的动机驱动的认知对话控制模型之前，我们先给出本文研究的认知对话系统（以下简称“CogDial”）的概念模型和框架。目前该系统只能处理英文对话，接下来章节中的对话的例子基本按照英文的习惯举的例子，可能不太适合中文的习惯，但是该系统框架中所采用的技术和理论基础都是完全适合中文的。

简单地说，CogDial的系统流程可以描述如下，此系统将接收到的人类话语转化成基于超图的抽象逻辑表达形式，概率逻辑网PLN会将这这些逻辑表达形式与系统的动机和情感状态，以及系统的知识库等结合起来，进行一定的解析和逻辑推理，从而得到想要回应的表达内容的逻辑表达形式，最后将这些逻辑表达式再转化成自然语言回应给谈话对象。概括地说，CogDial主要采用了以下几项技术：

* 基于动机驱动的情感模型Psi及其相关机制的宏观规划（Macroplanning）。该模块以言语行为理论为指导，主要用于以下两方面内容：1）在哪个时间点发生哪些言语行为；2）使用哪些内容去封装这些言语行为。宏观规划更多的是从语用和推理方面出发，淡化语义或语法（或者音韵词汇）等的限制。
* 基于概率逻辑网络的概率逻辑推理技术，用于实现Psi上的动机和行为的自适应规范化
* 将自然语言语句通过句法分析和语义逻辑关系抽取转换成基于超图的抽象逻辑形式表示的自然语言理解技术（这将在本文的第五章中介绍）。
* 将经过宏观规划得到的以超图形式表示的要表达的抽象内容进行微观规划（Microplanning）和表层生成技术最终转换成自然语言语句（这将在本文的第六章中介绍）。
* 在实现模式上，我们将上述两模块都封装在“言语行为规划器（Speech Act Schema）”中，以Context∧Procedure→Goal的蕴涵式形式表示，这些言语行为规划器将用于：
  + 在动机驱动的情感模型控制机制下，结合当前的语境，激活相应的言语行为规划器
  + 收集相关内容，生成能与微观规划衔接的Atoms集合
  + 调用一系列的认知机制（主要包括用于概率逻辑推理机制PLN和注意力经济分配机制ECAN [96]），选择相关的Atoms送到微观规划器中用于表层生成自然语言语句。

CogDial的初期目标主要是实现面向游戏角色和人形机器人的对话系统，但同样的研究思路也完全能应用在基于文本的对话系统，例如智能对话搜索接口。我们分阶段来实现这样的系统，以一个相对小而简单的系统为起点，通过不断改进和完善，以及结合系统本身的自学习和认知能力，最终实现一个在一定程度上接近人类水平的系统。目前，我们的系统还没有达到最终的水平，但基本框架已经到位，并用在汉森机器人公司的类人机器人Sophia[[3]](#footnote-4)上 。

## 动机驱动的认知对话系统模型

XXX

In our cognitive approach to dialogue, dialogue control occurs via the combination of multiple cognitive processes on a shared set of knowledge representations. The dynamics of dialogue control is guided by a motivational system, based on Dorner's Psi model, that chooses actions and allocates system resources according to a set of goals defined over multiple time scales.

### 针对认知对话控制的Psi的配置

基于Psi的基本框架，我们选择了以下的具体需求用来指导我们的对话系统的行为：

* + - 社交需求（Aﬃliation）：与他人互动，希望被其他成员接纳的需求；取悦谈话对象可以视为此目标的一个特例
    - 能力需求（Competence）：通过对话达到某种目标的需求，衡量言语表达方式的指标
    - 确定性需求（Certainty）：智能体对自身语境的了解需求，特别是对目标的了解需求
    - 新颖性需求（Novelty）：维持智能的会话而不是简单重复的问答会话。

CogDial中的对话控制主要通过不同的需求来选择相应的言语行为规划器，因此，在选择言语行为规划器前，我们需要知道当前状态下上述每种需求的被满足的程度。鉴于目前的CogDial系统的有限能力，为了能更好地实现一个面向集成认知体系的智能会话系统框架，我们将上述的几项需求做了更具体化的定义：

* + - 社交需求（Aﬃliation）：我们对该需求进行了下面三种满足程度：
* 当对话系统正在与人或者其他智能体进行会话时，该需求在一定程度上被满足；
* 当系统正与多个人或智能体进行会话时，该需求被满足的程度提升；
* 当该系统的会话内容都属于积极向上的时候（通过情感分析技术实现），该需求被满足的程度达到最高。
  + - 能力需求（Competence）：此项需求需要通过OpenCog来评估。简单来说，对每一个目标，OpenCog记录着会话系统在过去完成该项目标的程度，然后根据当前的不同目标所占的权重，我们可以通过以下公式估算该需求被满足的程度（首先针对每一个目标，计算目标权重\*能达到该目标的概率，然后求总）。当然计算目标被完成的程度，还应该考虑实现目标的语境等因素，目前我们的系统更注重构建一个智能会话系统框架，因此在语境无关的假设下来衡量目标被实现的程度。
    - 确定性需求（Certainty）：如果会话系统正在和一个陌生人对话，或者系统无法理解大量被提及的单词或概念，那么当前的确定性需求被满足的程度就会降低。如果系统获取了新的可靠知识，那么该需求被满足的程度就会增加。
    - 新颖性需求（Novelty）：我们定义了以下几种方式来增加智能绘画系统对新鲜度需求的满足程度值：
* 多和不同的人类或智能体会话；
* 谈论新的话题或引入新的词汇和概念；
* 学习新的可靠信息（OpenCog推理得出的新的置信度高的知识存储的载体原子（Atoms））

基于上述目标需求的智能会话系统，除了需要结合前文所述的OpenPsi的框架理论，以及本文描述的自然语言理解和生成的相关工具之外，还需要有以下模块：

* + - 制定一组能被特定目标需求激活的对话控制程序
    - 建立目标和相应去实现目标的行为之间的关联，可通过相关规则来实现，也可以通过强化学习来实现，我们系统框架采用两种结合的方法，但目前的系统还是以规则关联为主。

4.2.2 动机驱动的认知对话系统中的行为选择机制

正如Stan Franklin[105]指出的，智能的智能体的本质在于它可以做事，还可以采取行为。我们的认知对话系统CogDial中的行为选择会涉及OpenCog中所采用的行为选择机制[106]，因此本章节首先简单介绍该行为选择机制的基本思路，以及Psi的变体模型如何在OpenCog中处理动机（包括情感、驱动、目标等）和对行为选择的指导。

OpenCog中的行为选择机制的关键部分在于：

* + - 行为选择器（Action Selector）根据当前环境选择可能帮助实现重要目标的程序
* 例如：假设当前十分重要的目标是取悦谈话对象，一旦谈话对象问了问题，那么回答该问题会被评估为很可能可以达到“取悦谈话对象”的目标的程序。
  + - 为了支持行为选择器，OpenCog设立了这样的蕴涵式Context&Procedure →Goal，其中上下文（Context）是一个被赋值于智能体处境的谓词，在智能对话系统中，我们视为语境。
* 例如：如果Bob请求智能体（认知对话系统）去写一段话，又假设该智能体知道Bob非常执着，那么该蕴涵式可以被用成：

\* “Bob命令执行X”and“执行X”→“取悦Bob”<.9,.9>

* 例如：如果智能体想说服谈话对象相信某个陈述X，那么该蕴涵式可以被用成：

\* “告诉Bob我为什么相信X是正确”and“我想要说服Bob相信X是正确”→“满足度”<.9,.9>

* + - 这些蕴涵式的真值可以根据经验和推理来赋值。
* 例如：第一个例子中的蕴涵式的真值可以根据经验来赋值，即通过与记忆里Bob给出指令相关的情节来判断
* 例如：也能根据推理来赋值，即根据与Bob相似的个体给出指令的经验来类推，或结合Bob本人的明确表达，和Bob的自我描述通常合理的知识来推断
  + - 重要值（Importance Value）可以通过经济注意力分配（Economic Attention Allocation）[106]在目标之间传播，推理可用于从现有目标中学习子目标。目前，这一点还未被利用在CogDial系统中，但我们相信这点能帮助实现更复杂的对话行为。
* 例如：如果Bob告诉智能体去做X，那么智能体将“取悦Bob”的目标分解为“做X”，那么“取悦Bob”的目标会将它的重要值分给“做X”的目标（同样，“做X”的目标会将其重要值分给它的子目标，或者分给执行“做X”）。
* Let *I*(*C*, *A*, *G*) denote the implication schematically denoted
* ImplicationLink
* ANDLink
* Context
* Schema
* Goal
* as used by OpenPsi. Then we may let
* *wI* = *s* ∗ *c*（4-1）
* where *s* and *c* denote the strength and conﬁdence of the implication *I*; and let *S T IG* denote the ShortTermImportance (STI) of G.
* One may then deﬁne the weight of the schema *S* as
* *w* = *vS*

*S*

* ∑
* *R vR*
* （4-2）
* 其中 *i i i*
* More complex forms are also interesting, e.g.
* *Ii*(*Ci*,*S* ,*Gi*)
* （4-4）
* where *S T I*(*I*) refers to the STI of the implication I, and *w*(*Gi*) refers to the general, per- sistent importance of the goal *Gi* to that particular system (which is diﬀerent than the short-term importance of the goal to the system at that moment).

### 动机驱动的对话控制模型

The key steps in our Psi-based dialogue control process are as follows:

1. Perception
2. Evaluation of urges and desires
3. Evaluation of modulator values
4. Formulation of the current goals
5. Formulation of currently available actions
6. Action selection
7. Action orchestration

For the purpose of simple exposition, we will now step through these 7 steps in sequential order and explain how they come together to enable intelligent, motivation-driven responses in a dialogue context. However, we must emphasize that the actual dynamics in our model are not unidirectional. Alongside the feedforward dynamics in which perceptions trigger actions, the framework also allows for feedback dynamics in which the process of formulating actions guides perception. Diagram XX shows both the feedforward and feedback information flows involved in our model.

**Step 1. Perception**

Perception in our framework involves mapping linguistic and nonlinguistic perceptions into a common logic-based knowledge representation, which is the same one used to represent system goals and actions, and general background knowledge. In our conceptual model of dialogue control, we allow for cognitive dynamics to influence the nature of the mapping from sentences, or from e.g. visual perceptions, into logical representations. In our current prototype implementations, the transformation of linguistic surface forms and visual camera input into logical representations have been primarily feedforward and rule-based.

**Step 2. Evaluation of Urges and Desires**

Evaluation of Urges and Desires, involves the dialogue control system estimating, at each point in time, the degree to which each of its top-level goals (Ubergoals) and explicitly represented subgoals is fulfilled. For some goals, this will depend only on various predicates evaluated based on the system's current situation. For other goals, this will also depend on internal variables associated with the particular goals (e.g. in a dialogue system with a high motivation to seek novelty, after it has not experienced any novelty for a certain period of time, its novelty Urge may gradually increase in value, representing a steadily increasing boredom).

The derivation of subgoals for the system's top-level Ubergoals is carried out by probabilistic logical inference. Often this inference will occur via system background processing, independently of the systems real-time dialogue-control behavior. However, in cases where complex and adaptive discourse planning is required (e.g. a debate against a crafty opponent) then inference can also be used in real-time to create new subgoals, which then have their degree of fulfillment evaluated alongside the base-level Urges and Desires and the higher-level goals.

**Step 3. Evaluation of Modulator Values**

(Evaluation of Modulator Values) involves evaluation of the modulator parameters that, in accordance with the Psi model, guide the particulars of the system's action selection and action response.

In our prototyping work, we have evaluated the core Psi modulator values as follows:

* Valence: Recognition of positive and negative words and phrases in what the human dialogue participant says, and recognition of positive and negative tones in their voice (where the latter is determined by supervised learning models trained based on annotated data).
* Arousal:
  + Verbal communication with the dialogue system elicits arousal
  + Acoustic or verbal cues indicating urgency, or intensity of emotion, will elicit increased arousal
* Resolution level, in our prototyping work, has been left constant or set inversely proportional to arousal.
* Selection threshold:
  + The default selection threshold is a "personality parameter" which controls how "scatterbrained" the dialogue system appears
  + This parameter can also be adapted, so that e.g. the selection threshold is higher (and the system thus appears more persistent and single-focused) if it is detected that the human dialogue participant is especially serious in orientation in the particular interactions
* Goal-directedness:
  + In our experimentation so far, if the human dialogue participant is asking complex questions, then the goal-directedness is high; and if not, then it is assumed that more of a “casual conversation” mode is in place and the goal-directedness is lower.
  + In future work goal-directedness will need to be varied as part of dialogue planning
* Securing rate (frequency of background checks):
  + Some novel salient event occurring in the non-linguistic environment should increase the securing rate.
  + Being engaged in a conversation should decrease the securing rate

In our prototyping work, we have generally worked with a relatively limited set of Ubergoals as appropriate for simple dialogic interaction, e.g.

* affiliation (positive social interaction), which has been measured in two ways
  + via a default assumption that any dialogue interaction in which some human responds to the system provides positive “affiliation”
  + via emotion recognition from text, facial expression and voice, so that positive emotional responses on the part of the human dialogue participant is perceived to provide an amount of extra “affiliation”
* novelty (exploration),
  + which has been measured using the probabilistic truth values of the logic expressions in our system's knowledge base (using the grounding of the psychological concept of “novelty” in the mathematics of information theory, according to which a dramatic shift in the truth value of a logical predicate may be considered as “novel”).
* knowledge (competence)
  + which is measured via the probabilistic truth value of new logical relationships entered into the system's knowledge base
* aesthetics
  + which we have estimated in a very rough way based on the variety of emotions expressed by the human dialogue participant, according to the simple assumption that emotional expression is intrinsically aesthetically positive to a small degree

**Step 4. Formulation of the Current Goals**

(Formulation of the Current Goals) involves a weighting aspect, and a more creative and adaptive inference aspect. In the simplest cases, what is involved is just assigning weights to various system goals based on the current modulator values and any internal dynamics associated with the goals. However, in cases where more complex dialogue planning is required, then subgoals will need to be synthesized adaptively in response to the current context, via uncertain inference; the assignment of weights to these subgoals is then done as part of the subgoal learning process.

**Step 5. Formulation of Currently Available Actions**

(Formulation of Currently Available Actions) again involves a weighting aspect, and a more creative and adaptive inference aspect. In the simplest cases, what is involved is looking in the system's knowledge base for implications of the conceptual form CONTEXT and ACTION implies GOAL (which we call “Psi implications), and then weighting each such implication according to the current situation (according to mathematics we have outlined in Section 4.2.2 above). But in cases where more complex dialogue planning is required, then new implications of this form will need to be synthesized based on available knowledge.

In our current prototyping work, we have experimented with dynamic synthesis of such implications in a few situations, for instance when information about user state is only indirectly given and needs to be inferred. Primarily, however, we have relied upon Psi implications that correspond to commonly recognized speech act types, drawn from the SWBD-DAMSL speech act ontology as reviewed in Section 1.2.4 above. Via hand-coding Psi implications corresponding to different speech act types, we are able to create a dialogue system covering various everyday human interactions. This system then records its experiences in its logical knowledge base, and recognizes patterns in its experience, enabling it to ongoingly synthesize new Psi implications to guide its future actions.

XXX there are some specific examples from the previous dialogue-system draft that may fit in here XXXX

**Step 6. Action selection**

Step 6 (Action selection) occurs according to our adaptation of the Psi model, as we have outlined in Section 3.3 above. The logical implications gathered in Step 5 are then evaluated, and actions are stochastically chosen based on multiple factors. Actions involving language production are executed via invocation of the microplanner, which in turn invokes the surface realizer.

**Step 7. Action Orchestration**

Step 7 (Action orchestration) handles mediation between multiple potentially contradictory system actions. For instance, in a multi-user dialogue context, the system might be answering person A's question with a lengthy answer, and then person B interrupts with a simple Yes-No question. The action selector may suggest a “Yes” response to person B, but the job of deciding when to interrupt the response to A to utter this “Yes”, is left to the Action Orchestration process. In many cases some feedback between action orchestration, action formulation and action selection will be required, for truly flexible and intelligent dialogue behavior.

### 基于概率逻辑推理的语篇管理模型

Since the central role in this dialogue control dynamic is played by the system's goals and motivations, the achievement of complex dialogue management using this framework relies on the balancing of goals with multiple time-frames. For instance, if the dialogue system is interacting with person A and has a high-level goal of gathering information about A, it may nevertheless not be intelligent for the system to pursue this high-level goal directly via every utterance it makes. The system must be able to, for example, maintain “gather information about A” as a goal to be pursued over a 30 minute or one hour time-frame, but then be able to prioritize “make A happy” over “gather information about A” on a one-minute time-frame. In this case we simply have Psi implications that are making implications on different time scales, and the goal and action formulation phases of the dialogue process must be able to accommodate this (as they are, even in our current prototype system).

PLN as a logic system incorporates temporal reasoning, and is able to carry out planning and reasoning as parts of the same integrated cognitive process (XX). Real-world everyday dialogue structure also contains many “human” elements going beyond logic formalism, e.g. conventional narrative structures (XX); however, we do not attempt to encode these elements explicitly into our system, but rather endeavor that the system may learn these via uncertain inference based on its own experience.

## 言语规划器

XXX开场白

### 言语行为规划器

XXXX 这章可能需要调整。针对不同言语行为类型实现相应的言语行为规划 器，本节主要讨论如何在每个言语行为类型中使用情感计算模型和逻辑推理等进行 相应的规划

“言语行为规划器”（Speech Act Schema）是CogDial的一个核心设计，每个言语行为规划器里包含一种特定的言语行为（Speech Act），以及与该言语行为对应的特定的认知行为（Cognitive Procedure）。每种言语行为规划器都能在不同情况下被激发调用。对于言语行为规划器的选择，我们采用OpenCog中的OpenPsi的动机驱动模块来执行。

CogDial的总体规划是针对本文第1章综述里提到的42种SWBD-DAMSL言语行为[89]，实现相应的言语行为规划器。CogDial目前只实现了42种中的部分言语行为对应的言语行为规划器。另一方面，CogDial中的言语行为规划器的设计也绝不局限于这42种言语行为，CogDial实现了一些言语行为规划器能对应这42种中的某两种或多种言语行为，比如我们利用真值回答规划器（TRUTH VALUE ANSWER schema）来同时关联其中的YES QUESTION和NO QUESTION，从而可以将一般疑问句的回答根据真值的大小延伸到“可能”“不确定”“可能不”等，而不仅仅是局限于回答“是”或者“不是”。CogDial还根据智能体的具体情况拆分了这42种中的某些言语行为，例如，我们针对言语行为“STATEMENT”实现了两个不同的言语行为规划器：回应声明以及智能体自发的用于表达其自身状态和想法等的声明。

总的来说，本文虽然没有完全照搬SWBD-DAMSL的言语行为分类，但是，该分类体系是来源于对大量人类会话的进行具体分析后的实验结果，也在具体的会话分析和抽象的言语行为理论之间架起了桥梁，使得抽象的言语行为理论在机器上实现变得可行，因此，SWBD-DAMSL的言语行为分类体系还是很有借鉴价值的。

要实现上述的言语行为规划器，每一个言语行为都会引发一个相应的认知行为，也就是说，每一个言语行为都会触发调用一段程序；而这样的机制正好符合OpenCog里的GroundedSchemaNode的用法。GroundedSchemaNode是OpenCog的超图知识库里的一种节点类型，通常被封装在ExecutionOutputLink里，连接着一段代码（一般情况下是Scheme或者Python编写的代码）的名称。该代码可以通过ExecutionOutputLink被触发和执行。因此，鉴于这样的执行机制，我们可以通过对每个言语行为规划器定义一个GroundedSchemaNode来实现言语行为规划器。这样的设计方案可以减少冗长复杂的代码块，而直接通过OpenCog中统一又通用的Atomspace的基本操作方法来管理和执行复杂的言语行为规划器。另外，这样的机制也能很容易调用知识库Atomspace之外的复杂程序，从而得到很好的扩展性，也为言语行为规划器的扩展研究搭建了个很好的平台。例如，我们可以通过调用外部程序将逻辑推理系统PLN的前向或者后向推理应用到言语行为规划器中，使其能实现自适应学习，不断完善言语行为规划器。本文后面会进一步讨论这样的扩展。每一个言语行为规划器的输入是一个被叫做对话节点（DialogueNode），DialogueNode是可以表示一方或者多方之间的会话交互的节点类型。DialogueNode可以有不同话语节点（UtteranceNodes）作为成员，在实现方式上，DialgoueNode通过MemberLinks指向不同的UtteranceNodes。而UtteranceNode则可以关联以下不同类型的节点：

* + - 关联一个或多个文本节点（TextNode）、句子节点（SentenceNode）或者短语节点（PhraseNode），则表示输入的话语内容来自一个或者多个文本、句子或者短语。
    - 关联一个声音节点（SoundNode），则表示该话语来自外界声音。
    - 关联指向说话者的Link。
    - 关联指向对话语的补充信息的Links，这些补充信息可以是该话语的言语行为类型、与该话语相联系的情感等。
    - 关联指向产生话语的言语行为规划器，用于响应是什么触发该话语。
    - 关联一个或多个解析节点（InterpretationNode），这些解析节点可以是用来解析话语的语义和语用信息。

言语行为规划器的输出是连接了一系列相关联Atoms的SetLink，该输出会送入本文第6章中描述的微观规划和表层生成等工具，从而产生相应的自然语言来回应输入的内容。言语行为规划器还会将输出的话语关联到产生该话语的DialogueNode，这样不仅仅是记录了会话的内容，还能用于智能体的强化学习和提高智能体的各种需求的满足度。下面通过一个例子来解释上述的实现过程。假设有下面的简单对话：

Ruiting： How are you doing?

CogDial： I am fine

这个简单的对话可用以下的Atoms来表示：

MemberLink

UtteranceNode [555]

DialogueNode [123]

MemberLink

UtteranceNode [666]

DialogueNode [123]

EvaluationLink

PredicateNode "say"

ConceptNode "Ruiting"

UtteranceNode [555]

EvaluationLink

PredicateNode "say"

ConceptNode "me"

UtteranceNode [666]

EvaluationLink

PredicateNode "Textual Content"

UtteranceNode [555]

SentenceNode "How are you doing?"

EvaluationLink

PredicateNode "Textual Content"

UtteranceNode [555]

SentenceNode "I am fine."

EvaluationLink

PredicateNode "Utterance Type"

UtteranceNode [555]

ConceptNode "Interrogative"

EvaluationLink

PredicateNode "Utterance Type"

UtteranceNode [666]

ConceptNode "Declarative"

AtTimeLink

UtteranceNode [555]

TimeNode "22：15：33 12/06/2014"

AtTimeLink

UtteranceNode [666]

TimeNode "22：15：47 12/06/2014"

EvaluationLink

PredicateNode "Interpretation"

UtteranceNode [555]

InterpretationNode [22]

EvaluationLink

PredicateNode "Interpretation"

UtteranceNode [666]

InterpretationNode [33]

MemberLink

EvaluationLink

PredicateNode "doing"

ListLink

ConceptNode "you"

VariableNode "var1"

InterpretationNode [22]

MemberLink

InheritanceLink

ConceptNode "I"

ConceptNode "fine"

InterpretationNode [33]

ExecutionLink

GroundedSchemaNode "polite\_banter.scm"

ListLink

UttteranceNode [555]

DialogueNode [123]

UtteranceNode [666]

其中的命题也可以有多种选择，例如：

EvaluationLink

PredicateNode "Conversation Partner"

DialogueNode $D

$X

该命题为真，当且仅当，X是D其中一个话语的发言人。

### 言语行为规划器及其关联的目标和上下文的实现

在[89],中，Twitchell和Nunamaker根据Searl的言语行为理论的经典分类，在对大量的人类会话进行经验分析后，将言语行为细分为42种。虽然此分类体系很有理论研究价值，但在实验过程中，考虑到实用智能对话系统的语境等因素，我们对这42种言语行为做了稍微调整，同时也在CogDial中添加了一些SWBD-DAMSL研究中没有出现言语行为。

即使在有明确的言语行为类型的情况下，仍然有很多种方法去构建一个智能对话系统。CogDial根据多个广泛的可扩展目标制定一些特定的设计决策，在这些决策基础上，系统能通过自适应学习方法自动改进，为能跳出传统的对话系统的研究方法搭建一个基础平台。为了搭建这样的系统，需要考虑的第一点是，对于每一个言语行为，都有相应的固定形式的认知内容被触发，也有相应的习惯性表达方式来表达该认知内容。

如果在类似的言语行为类型分类基础上构建一个简单的“聊天机器人”，一般通过更简单，不用加入很多认知处理，针对每个言语行为类型，输出直接的具体的话语，也同样能达到类似的效果。例如，对于言语行为类型Agree，可以编写简单的程序使智能体输出“同意（Agree）”或者“是的，我同意（Yes,I agree)”。目前，“聊天机器人”的概念很模糊。我们这里提到的“聊天机器人”范围也很广，可以是完全基于模板匹配的聊天机器人ELIZA[2]，也可以是具有一些推理能力但是基本忽略语义理解的众所周知的Siri。但关键的一点是，我们设计的CogDial系统，是本着该系统能“理解”自己在说什么的理念，也就是说，该系统所产生的话语来自于内部的语义关系图，且该语义关系图和系统知识库中的其他语义关系图之间有着丰富的语义关系。我们希望系统能在一定程度上更深地“理解”自己在说什么，而不是只生成它不理解的字符串。在某些情况下，在不理解的情况下破口而出一些话语，尤其是习惯用语，也是可以接受的（其实人类有时候也会无意识地这么做），但这并不是大部分情况。

在CogDial系统的设计中，每一个言语行为都需要下列几项内容与之相应：

* + - 相应的目标和语境二元组（goal，context），表明何时该言语行为会被触发。
    - 相应的能生成相关信息的程序，产生由该言语行为引起的要传递给谈话对象的信息。
    - 一个或多个相应的“语义模板”，表明由该言语行为引发的相关认知内容。
    - 连接上述语义模板所在的Atoms和特定的句子实现的Atoms的Links，用于传送到Microplanning，从而生成相应的话语。

这样一来，通过编写一些抽象的语义形式和特定的会话习惯之间的匹配模式，便能构建出一个具有一定合理功能的认知对话系统，当系统学习了用不同的更复杂的表达方式去实现抽象的语义形式后，也就能在更大程度上“理解”会话内容。此外，这些抽象的语义形式除了与言语行为和目标需求相关联，还和其他不同的认知内容相关联，因此，系统会随着经验的增长而趋向成熟。假设任何言语行为被触发都能增加下面表达式中的EvaluationLink的真值：

EvaluationLink

PredicateNode "Currently Having Conversation"

TimeNode T

其中，“T”表示当前时间。又假设上面表达式中的EvaluationLink和系统中多个顶层目标有关联（该假设对于某些应用也不总为真，那么在这样的情况下，这些不为真的关联的链的权重会根据具体情况被调为适当的值），也就是说，该系统能看到会话行为带来的价值。因为每个言语行为都蕴涵着这个EvaluationLink，所以每个言语行为都会影响系统的目标实现。一些言语行为会通过持续进行对话从而超标完成某个系统目标。这样的言语行为会在后面章节详述。下面会给出一些具体的例子进一步解析前面几段提到的Atoms。

ImplicationLink <.5>

EvaluationLink

PredicateNode "Currently Having Conversation"

TimeNode $T

EvaluationLink

PredicateNode "Increase Knowledge"

TimeNode $T

上述超图片段表明，维持对话能在一定程度上满足系统目标“增长知识”，ImplicationLink的初始权重设为0.5，表明“当前有会话”蕴涵系统目标“增长知识”只有0.5的概率。这个权重会随着系统的经验而改变，也会通过其关联的其他节点和链的具体情况推算得来。例如：

ImplicationLink <.1>

ANDLink

EvaluationLink

PredicateNode "Currently Having Conversation"

TimeNode $T

EvaluationLink

EvaluationLink "DialoguePointer"

PredicateNode "Currently Having Conversation"

DialogueNode $D

EvaluationLink

PredicateNode "Conversation Partner"

ConceptNode "Bob"

EvaluationLink

PredicateNode "Increase Knowledge"

TimeNode $T

上面的超图片段表明，当进行对话的对象是Bob的时候，只有0.1的概率能实现系统目标“增长知识”。

ImplicationLink <1>

ExistsLink $G, $S, $O, $U, $D

ANDLink

MemberLink

GroundedSchemaNode $G

ConceptNode "Speech Act Schema"

AtTimeLink

TimeNode $T

ExecutionLink

GroundedSchemaNode $G

$S

$O

MemberLink

UtteranceNode $U

DialogueNode $D

EvaluationLink

PredicateNode "Textual Content"

UtteranceNode $U

SentenceNode $O

AtTimeLink

TimeNode $T

EvaluationLink

PredicateNode "Currently active"

DialogueNode $D

EvaluationLink

PredicateNode "Currently Having Conversation"

TimeNode $T

上面的例子说明，如果一个言语行为规划器被执行，当前的对话节点（DialogueNode）D会关联一个谓词“当前活跃”，表明D出于活跃状态，那么“当前有会话”的目标被实现。

MemberLink

GroundedSchemaNode "answer\_yes.scm"

ConceptNode "Speech Act Schema"

接下来的例子演示了言语行为如何与目标关联：

ImplicationLink <.8>

ExistsLink $S, $O, $U, $D

ANDLink

AtTimeLink

TimeNode $T

ExecutionLink

GroundedSchemaNode "answer\_yes.scm"

$S

$O

MemberLink

UtteranceNode $U

DialogueNode $D

EvaluationLink

PredicateNode "Textual Content"

UtteranceNode $U

SentenceNode $O

AtTimeLink

TimeNode $T

EvaluationLink

PredicateNode "Currently active"

DialogueNode $D

EvaluationLink

PredicateNode "Please Conversation Partner"

TimeNode $T

上面的例子表明言语行为规划器“肯定回答”（”Answer Yes”schemma）可以用于增加实现“取悦谈话对象”目标的幅度值，超越了谈话对象因单纯继续对话而被取悦的程度。

许多言语行为规划器都会有不同的清晰表达相关语义内容的方式。比如说，一般会话的开头，可以说“你最近状态怎么样？”（”What has your state been recently?”），或者“你最近都忙什么？”(“What have you been doing?”)等，而类似这样的语义内容很容易约定俗成地被说成“最近怎么样？”（“What’s up?”)。对于机器来讲，如果对话系统直接问“What’s up?”当然也没问题，但是有必要使系统知道“What’s up?”只是其他两种说法或者“what are you thinking about?“的一种约定俗成的简约说法。系统会根据智能体的不同个性对每种不同的说法一个相应的权值。

一般来说，会存在很多的“个性参数”影响着多种言语行为。在CogDial的实现过程，我们针对那些对对话影响较大的关键特性（我们称为“对话特征”（Dialogue-trait））创建相应的概念节点（ConceptNode），比如：习惯用语（Idiomaticity）、非正式（Informality）、精确（Precision）、累赘（Wordiness）、开放（Openness）。用来表示由言语行为规划器产生的具体话语的SetLink，将以不同的权重与这些表示不同对话特征的ConceptNode相连。例如，表示“I dunno”的SetLink将以较高的权重与“Informality”以及“Idiomaticity”关联，以较低的权重与“Precision”“Wordiness”关联。这些对话特征的参数可以在对CogDial设置基本参数的时候根据不同的需求人为设计。也可以在对话过程中根据谈话对象的喜好来自适应地调整。

综上所述CogDial的整个设计方案中，需要人工参与的部分包括：

* + - 自然语言理解流水线中的提到的规则（最终会被我们正在研究的无监督语言学习所取代，本文第9章有更详细的阐述）
    - OpenPsi中的顶层目标
    - 不同的言语行为所引发的不同认知过程。目前这些过程是在与GroundedSche-maNode绑定的相应Scheme或者Python代码中被实现。这些认知过程也可以在OpenCog的知识库Atomspace里被实现（如下文中的Question-answering schema）。

本章接下来的部分将进一步阐述CogDial使用的一系列言语行为规划器以及解释它们在CogDial中的实现机制，这些言语行为规划器大部分从SWBD-DAMSL中借鉴。当然这些特殊的言语行为规划器的集合并非一成不变，在以后对系统的不断改进和完善过程中，无疑会导致对该集合一定程度的延伸和细化。但我们相信这是一个好的开始，需要重申的是这些言语行为规划器分类是大量人类对话的实证分析的结果。目前，我们的研究工作重点在问答规划器（Question-answering schema），将会在第八章中进一步阐述。

**1. 谈话开场白、结束谈话和转移话题**

* 被放弃或转移、结束
* 例如：“所以，嗯......”（“So,hmmmm....”）
* 相关目标、语境：当前谈话不尽能达到智能体的目的；但其中一些与谈话对象的对话，似乎仍有达到智能体目的合理程度上的可能性。或者，智能体无法想到任何与谈话对象所讲之相关的答复。当持续对话被断定为可达到智能体的某些目的，然而其他言语行为似乎无法在显着程度上充分达到智能体之目的；或当言语行为极有可能达到智能体的目标，然而看似与先前的对话失去连结性，在这样的情况下，这就会被使用（因此，某些言语标志对于划定新的谈话阶段之界线，是很恰当的）。
* 程序：在此情况下，待给予的语意内容往往会是”或许我们该来谈点别的”(”Perhaps we should talk about something else”)、或”现在来谈谈别的吧。”(”Let us now talk about someone else”)这样的语意内容会搭配清晰的咬字，也涉及社交上常见的言辞，比如“嗯......这个嘛......”(“Hmmmm...welll...”)。
* 常见谈话开场白
* 例如：”最近过得怎样？”(”How’s it going?”)
* 相关目标、语境：一名潜在交谈对象在现场，经断定，与该对象交谈将能达到系统的目标。
* 程序：”常见谈话开场白”的性质，会以表示问候、欲进行交谈的一种社会成规作为开端。有些常见的谈话开场白非常普通，例如”嗨。”(”Hi.”)而其他较有语意内涵的，比如”现况如何？”(”What is your current state?”)、”最近经历了哪些事？”(”What have been your recent experiences?”)、”在想些什么？”(”What are you thinking about?”)、”目前在从事些什么？”(”What are you involved with currently?”)这样的语意内容会搭配清晰的咬字，也涉及社交上常见的言辞，比如”最近怎样？”(”What’s up?”)、”有什么新鲜事？”(”What’s new?”)、”生活怎么样？”(”How’s it going?”)等。同样地，也可能会明白地问、或通过惯用语表达”我想和你谈谈”(”I would like to talk to you”)或”想聊天吗？”(”Would you like to chat?”)等语意内容。
* 结束谈话
* 例如：”那好吧……今天跟你聊天很开心。”(”Alrighty then...it’s been good to chat with you.”)
* 相关目标、语境：如结束谈话会是达成系统目标的最佳方式，这样的话语是很适当的。若确定谈话对象欲结束谈话，在这种情况下，结束谈话会是达成系统目标取悦谈话对象最好的办法。在任何情况下，比起唐突结束，用言辞结束谈话反而是达成取悦人类的目标最佳的方式。
* 程序：结束谈话的语意内容会有”这次谈话我很尽兴”(”I have enjoyed the conversation”)、”谢谢你给我这么有意义的谈话”(”Thanks you for the good conversation”)、”希望下次还能再跟你聊天”(”I hope to talk to you again”)等，并且是以直接或惯用性言辞表达。

**2. 实质性对话开场白**

在CogDial系统的语境中，有许多种对话开场白往往很实用，但这并没有在SWBD-DAMSL研究中被提出讨论。例如：

* 意识流陈述性开场白
* 例如：“我常在想，有些人总在谈些同样的事。”(“I’ve been thinking there are some people who always talk about the same things.”)
* 相关目标、语境：为取悦谈话对象、或达到令人出奇的目标，就会通过此等开场白达成。
* 程序：在此开场白的程序，会先从AttentionalFocus截取一组Atom，再将之供给微规划程序进行发音。
* 个人化开场白
* 例如：“我对下届的总统大选有些看法。”(“I have some thoughts about who’s going to win the next Presidential election.”)用于和时常谈论政治的人谈话的场合
* 相关目标、语境：与意识流陈述性开场白相同，但更着重于取悦谈话对象和增进联系。
* 程序：对代表谈话对象的Atom予以高度的重视值（在OpenCog术语中的ShortTermImportance），接着在其散播到AtomSpace一段时间后，从AttentionalFocus中选择一组Atom，再将之供给微规划程序进行发音。

从某种意义上来看，这些言辞只是陈述句；但事实上，它们被用作对话开场白，使之增添了些许不同于平常的韵味。不同的认知程序经常会被用在选择哪些语句该作为对话开场白。

在有些相关言语行为中，问句会被用作对话开场白。例如：“谁会赢得下届的总统大选？”(“Who’s going to win the next Presidential election?”)这些都可为如下探讨的任何问句形式。但演算出该问什么来开始一段对话，与演算出该用何种语句来作对话开场白的程序，会有高度的相同性。

**3. 常见反应**

* 了解（衬托型反馈形式）
* 例如：“嗯，了解。”(“Yep,understood.”)
* 相关目标、语境：此达成了取悦谈话对象的目标（因为多数人都喜欢谈话时被了解的感受）；由于表示了解某谈话要点，使得谈话对象似乎可继续传达下个谈话重点，因此也可达到增进新奇感和知识的目的。基本的语境条件，在于谈话对象说了某些智能体了解的话。另一方面，选择此规划器的重要性在此情形下会更高：谈话对象不太确定智能体是否了解，或者谈话对象似乎会重复相同的信息（例如智能体连续给予谈话对象两个带有高度相似内涵的言辞）。若智能体强烈了解谈话对象的表达而胜过同意之，这时选择此规划器的重要性也较高。
* 程序：语意内容如”我了解你刚所说的”(”I understand what you just said”)；这般言辞可明确或以惯用语方式传达。
* 同意、接受
* 例如：”你说对了。”(”You got it.”)
* 相关目标、语境：此达成了取悦谈话对象的目标（因为多数人都喜欢谈话时被了解的感受）；由于表示了解某谈话要点，使得谈话对象似乎可继续传达下个谈话重点，因此也可达到增进新奇感和知识的目的。基本的语境条件，在于谈话对象说了某些智能体了解、并且同意的话。另一方面，选择此规划器的重要性在此情形下会更高：谈话对象不太确定智能体是否了解，或者谈话对象似乎会重复相同的信息（例如智能体连续给予谈话对象两个带有高度相似内涵的言辞）。
* 程序：语意内容如”我同意你刚所说的”(”I agree with what you just said”)；这般言辞可明确或以惯用语方式传递。
* 欣赏
* 例如：”是啊，我很确定……”(”Yeah,I’m sure....”)
* 相关目标、语境：此达成了取悦谈话对象的目标（因为多数人都喜欢谈话时被了解的感受）；由于表示了解某谈话要点，使得谈话对象似乎可继续传达下个谈话重点，因此也可达到增进新奇感和知识的目的。基本的语境条件，在于谈话对象说了某些智能体了解、同意、且满意的话。
* 程序：语意内容如”我很满意你刚所说的”(”I am caused pleasure by what you just said”)；这般言辞可明确或以惯用语方式传递。
* 对明白的回应
* 例如：”好的，明白你的意思了。”(”OK,gotcha.”)
* 相关目标、语境：这就如前述的”了解”行为，惟有基本的语境条件，在于谈话对象说了某些智能体了解、并回应某些智能体先前所说过的话。
* 程序：就如前述的”了解”情况，语意内容为”我明白你刚所说的”(”I understand what you just said”)；这般言辞可明确或以惯用语方式传递，但惯用语的表达方式会与”了解”的情况不尽相同。
* 重复措词
* 例如：”啊，你觉得他疯了。”(”Ah,you think he’s crazy.”)
* 相关目标、语境：此达成了取悦谈话对象的目标（因为多数人都喜欢谈话时被了解的感受）；由于表示了解某谈话要点，使得谈话对象似乎可继续传达下个谈话重点，因此也可达到增进新奇感和知识的目的。基本的语境条件，在于谈话对象说了某些智能体了解的话。另一方面，选择此规划器的重要性在此情形下会更高：谈话对象不太确定智能体是否了解。
* 程序：语意内容为重复谈话对象不久前所说的。首先，可重复谈话对象整体的言辞评论，或仅取其最重要的措辞句话。常见的言辞如”哦”(”Oh”)、或”啊？”(”huh?”)也可视情况添加。
* 道歉
* 例如：”对此我很抱歉。”(”Sorry about that.”)
* 相关目标、语境：此达成了取悦谈话对象的目标。主要的语境条件，在于谈话对象看似受到智能体所说、或未说的话之困扰或冒犯；其次较不重要的语境条件，在于谈话对象看似对其他事情困扰、不悦或受到冒犯。
* 程序：语意内容单纯为”我很抱歉”(”I am sorry”)或”对X我感到很抱歉”(”I am sorry about X”)；而“X”为对谈话对象造成负面反应的任何事物；这般言辞可明确或以惯用语方式传递。
* 道谢
* 例如：“太感谢你了，这真的很棒！”（“Thanks so much,that was fantastic!”）
* 相关目标、语境：此达成了取悦谈话对象的目标。主要的语境条件，在于智能体满意谈话对象所说的。另一个条件，在于”谢谢你”为社交上适宜的言辞，比如谈话伙伴明确称赞智能体的情况。程序：语意内容为”对X我很感谢”(”I am grateful for X”)，或单纯为”我很感激”(”I am grateful”)；这般言辞可明确或以惯用语方式传递。
* 置之度外
* 例如：”当然，没事的，别担心。”(”Sure,no problem,don’t worry about it.”)
* 相关目标、语境：此达成了取悦谈话对象的目标。主要的语境条件，在于谈话对象对于其所说的话或所做作为感到后悔；或对他自己说了些负面的话。
* 程序：语意内容为”对于X我并不深受其扰”(”I am not signiﬁcantly bothered by X”)或”那件事并没有太影响我”(”I am not signiﬁcantly bothered by that”)；这般言辞可明确或以惯用语方式传递。

### 4. 实质回应

* 协作完成
* 例如：“……没连任又担任了两届的总统”(”…who served two non-contiguous presidential terms”)[回应一句不完整的言辞”格罗弗•克利夫兰是美国唯一的一位……”(”Grover Cleveland was the only US president…”)]
* 相关目标、语境：此达成了取悦谈话对象的目标。主要的语境条件，在于谈话对象说了看似一句较长语句中一部分的话，而智能体正确猜到剩下的叙述为何。
* 程序：语意内容为智能体猜测谈话对象所说的其余叙述，完善一个句子。
* 引用
* 例如：”食言是很不可理喻的。”(”It doesn’t make sense to eat words.”)[被告知某人食言时的回应。]
* 相关目标、语境：例如当谈话对象发出的某些言辞，是智能体本身就有强烈评估的事物–这言辞显然正确、错误、令人讶异或令人开心等。
* 程序：语意内容为”P属于X”的形式；X为谈话对象先前发出的言论，而P为智能体强烈认定为X所拥有–例如事实、虚伪、惊讶、喜悦、不悦等。这种言辞的惯用语的表示并不多，但确实是有不少方式可表示此等言辞，像是”X使我高兴”(”X makes me happy”)、”X令我欣喜”(”X pleases me”)、”我喜欢甲”(”I like X”)之类。
* 总结/再阐述
* 例如：”所以说，你觉得他是个危险的疯子。”(”So you think he’s a dangerous madman.”)
* 相关目标、语境：此达成了取悦谈话对象的目标（因为多数人都喜欢谈话时被了解的感受）；由于表示了解某谈话要点，使得谈话对象似乎可继续传达下个谈话重点，因此也可达到增进新奇感和知识的目的。基本的语境条件，在于谈话对象说了某些智能体了解的话。另一方面，选择此规划器的重要性在此情形下会更高：谈话对象不太确定智能体是否理解、或智能体本身也不确定自己是否理解。
* 程序：语意内容为与谈话对象先前的言辞相同，或有时为谈话对象近来的一系列言辞。微规划系统会被特别要求找出不同表达此语意内容的方式，而不是重复谈话对象所说的话。
* 反问
* 例如：”若沙特阿拉伯不具备这些美国武器，那迪拜会有什么防御足以对抗贫穷非洲人的大举入侵？”(”What defense would Dubai have against an invasion by masses of impoverished Africans if Saudi Arabia didn’t have all those American weapons?”)
* 相关目标、语境：最基本的语境条件，在于有些疑问是智能体认为它对该问题思考过会较好。例如智能体认为某些问题是谈话对象会知道更多、或说出更有益的食物，而智能体对此问句更熟悉，在此情况就有可能发生；反问的问句便会附属这个语句。作用于此的主要目标是为了取悦谈话对象–而较间接、有把握（知识），因为当谈话对象了解越多，智能体也就会知道越多，这往往不会出错。
* 程序：此程序的关键在于确认是否有S的语句，若谈话对象知悉S、或对S有更深度的了解，谈话对象对当前谈话主题的知识会较渊博。若如此，以问句方式来构建S并提出这个问题，较有意义。
* 或者从句
* 例如：”还是说，他是为了自己而拿了所有钱？”(”Or did he take all the money for himself?”)
* 相关目标、语境：语境条件在于谈话对象发出带有内涵或清楚含义X的语句，但某Y排除了X，对智能体而言似乎也有合理程度上的强烈可能性（不必然，但或许与X的可能性同等强烈、或比X更强烈）。这部分达成的目标一般而言是取悦谈话对象，并增进知识。若证实提出的选项Y比选项X还要新奇，新奇感也会是相当重要的目标。
* 程序：此程序其实就是找出貌似极有理的Y，排除谈话对象表示的X。假定如此，待用言语表现的语意内容则为Y。
* 而且从句[[4]](#footnote-5)
* 例如：”而且他还自己吃光了所有奶酪？”(”And then he ate all the cheese himself?”)
* 相关目标、语境：语境条件在于谈话对象发出带有内涵或清楚含义X的语句，而智能体顺势地从X联想到某Y。这部分达成的目标一般而言是取悦谈话对象，并增进知识。若证实提出的选项Y令人惊讶，新奇感也会是相当重要的目标。
* 程序：此程序其实就是找出貌似极有理的Y，延伸谈话对象表示的X。假定如此，待用言语表现的语意内容则为Y。

**5. 回答**

* 肯定回答
* 例如：”是的，没错。”(”Yes,that’s right”)
* 相关目标、语境：语境条件在于谈话对象问了个是非问答，而智能体认为答案为肯定。目标是取悦谈话对象。
* 程序：语意内容为”是”(”Yes”)，措辞表达的方式有很多种。
* 否定回答
* 例如：”不，恐怕不是这样。”(”No,I’m afraid not.”)
* 相关目标、语境：语境条件在于谈话对象问了个是非问答，而智能体认为答案为否定。目标是取悦谈话对象（虽然在此情况下，达到目标的程度一般会比”肯定回答”来得稍微低些–比起否定的答复，多数人更想听到的是肯定回答，但这显然还是依特定情况而定）。
* 程序：语意内容为”否”(”No”)，措辞表达的方式有很多种。
* 拒绝
* 例如：”呃……不，我做不到。”(”Uh...no I can’t do that.”)
* 相关目标、语境：语境条件在于谈话对象提出了建议，而智能体认为这建议有误（若为陈述句）、不值得做或不可能办得到（若为命令句）。在拒绝某命令的情况下，此举是关系到所有智能体的目标，但仍取悦谈话对象和达成联系–因为一般来说，若智能体拒绝了某要求，是因从事智能体所想做的，会比从事谈话对象所请求的更能够帮助其达成其他目标。
* 程序：语意内容为”我不认同X”(”I don’t agree with X”)或”我不想从事X”(”I don’t intend to do X”)；措辞表达的方式有很多种，而X可被取代为”那个”或其他照应语。
* 肯定的委婉回答
* 例如：”对啊，她是这样。”(”Yes she did.”)
* 相关目标、语境：语境条件在于谈话对象问了个是非问答，而智能体认为答案为肯定。目标是取悦谈话对象。对此规划器应有成见的情况，在于智能体对问题的重视、或者谈话对象看似对该问题有特定程度的重视，正如这类回答带有特别的强调语气。
* 程序：一般程序为使语句S相应于谈话对象问的问题，并从该语句取一关键片段F，再以言语方式表达等同”我同意F”(”I agree with F”)、或”F是对的”(”F is true”)等措辞。
* 否定的委婉回答
* 例如：”这个嘛，我不认为……”(”Well I think not...”)
* 相关目标、语境：语境条件在于谈话对象问了个是非问答，而智能体认为答案为否定。目标是取悦谈话对象（虽然在此情况下，达到目标的程度一般会比”肯定回答”来得稍微低些）。对此规划器应有成见的情况，在于：问句或其答案看似有特别高度的重要性（OpenCog中的STI），正如这类回答带有特别的强调语气；或者，智能体对问题的重视、或谈话对象看似对该问题有高度的重视。
* 程序：一般程序为使语句S相应于谈话对象问的问题，并从该语句取一关键片段F，再以言语方式表达等同”我不赞同F”(”I disagree with F”)、或”F并不是对的”(”F is not true”)等措辞。
* 也许、部分接受
* 例如：”是啊–好像是吧。”(”Yeah–kind of.”)
* 相关目标、语境：语境条件在于谈话对象问了个是非问答，而智能体认为答案具有不太接近1、也不太接近0的真伪值。目标是取悦谈话对象。
* 程序：一般程序为使语句S相应于谈话对象问的问题，并从该语句取一关键片段F，再以言语方式表达等同”我部分同意F”(”I partly agree with F”)、”我认为F可能是对的”(”I think F is possibly true”)或”F有部分是对的”(”F is possibly true.”)等措辞。智能体会使用特定的惯用语表达特定程度的估计真实性，比如”我认为F也许是对的”(”I think F is probably true”)或”我觉得F有令人信服的真实度”(”I think F is conceivably true”)等。较动摇不定的讲法如”我估计F的可能性约为6，带有8的置信水平”(”I estimate the probability of F at approximately 6 with conﬁdence level.8”)（或被给予的任何数值）。
* 其他回答
* 例如：“我不知道。”（“I haven’t a clue.”）
* 相关目标、语境：语境条件在于谈话对象问了个是非问答，而智能体认为答案为肯定。目标是取悦谈话对象以及聚集知识。特定的语境条件，在于谈话对象问了个问题，而智能体也不知道答案、或对问题有其他反应，或问题的评估对智能体而言比该答案还要重要。
* 程序：此程序为对该问句识别出主观上的重要反应或评估。语意内容为”我有反应R”(”I have reaction R”)或”我对问题Q有反应R”(”I have reaction R to question Q.”)例如”我不知道”(”I have no idea”)、”我不知道是谁杀了J.R.”或”我真的很讨厌思考为什么人们如此邪恶”(”I really hate thinking about why people are so evil.”)。
* 不合意的回答
* 例如：”那其实不是我所想的。”(”That’s not really what comes to mind.”)
* 相关目标、语境：语境条件在于谈话对象回答了一个问题，但智能体认为答案虽行得通，但并不是最好的答案。目标是取悦谈话对象以及聚集知识。
* 程序：语意内容为“我认为X并不是Q最好的答案。”“I think X is not the best answer to Q.”，措辞表达的方式有很多种。

### 6. 问句

* 是非问答（稍微延伸SWBD-DAMSL；广义来说，如考量到答案为分等、而非是、否之二元真伪值的情况，则可将之视为真伪值疑问。）
* 例如：”你有给我编程吗？”(”Did you program me?”)
* 相关目标、语境：在此主要的目标，一般为获取知识和新奇感；而取悦谈话对象为其次。基本语境条件为：

1. 智能体或许某程度上重要的Atom，但置信度低；因此欲针对此Atom的真伪值提问。
2. 谈话对象或许提及了一特定的概念C，而C有几个层面是智能体知识较不足的；这可被确切阐述为真伪值疑问句。在此情况下，取悦谈话对象会是主要目标。

* 程序：将”Atom C的真伪值为何”(”What is the truth value of Atom C”)转换为一个句子，一般会从多种惯用语表达方式之中取一种。例如，智能体绝不应这么问：”猫吃老鼠的真伪值为何？”(”What is the truth value of cats eating mice?”)，取而代之，应这么问：”猫吃老鼠吗？”(”Do cats eat mice?”)；也不应这么说：”猪很笨的真伪值为何？”(”What is the truth value of pigs being stupid”)，取而代之，应这么说：”猪很笨吗？”(”Are pigs stupid?”)
* 特殊疑问句(Wh-Question)
* 例如：”谁创造了第一台电脑？”(”Who built theﬁrst computer?”)
* 相关目标、语境：在此主要的目标，一般为获取知识和新奇感；而取悦谈话对象为次要的目标。基本语境条件为：

1. 智能体或许带有可变因素的Atom，并且不知道此Atom的任何置信满意度，因此欲提问了解。
2. 谈话对象或许提及了一特定的概念C，而C有几个层面是智能体知识较不足的；这可轻易被阐述为真伪值疑问句。在此情况下，取悦谈话对象会是主要目标。

* 程序：
* 陈述式是非问答（真伪值疑问句）
* 例如：”所以你今天完全是走路去工作吗？”(”So you walked all the way to work today?”)
* 相关目标、语境：在此主要的目标，一般为获取知识和新奇感；而取悦谈话对象为其次。基本语境条件与一般真伪值疑问相同；但若智能体对问题的正确答案相当肯定（虽然并不是完全肯定），采用这个形式较为适当。
* 程序：语意内容为”X是正确的吗？”(”Is it correct that X?”)，但一般而言会以惯用语的方式表达。
* 疑问句形式的衬托型反馈
* 例如：”你确定？”(”Are you sure?”)
* 相关目标、语境：在此主要的目标，一般为获取知识和新奇感；而取悦谈话对象为其次。基本语境条件，在于谈话对象说了某些事S，而智能体认为S可能有误、或觉得S非常令人惊讶。
* 程序：语意内容为”你非常肯定S吗？”(”Are you highly certain that S?”)但一般而言会以惯用语的方式表达。
* 开放式问题
* 例如：”对于他的前途你有什么看法？”(”What do you think about his prospects?”)
* 相关目标、语境：

1. 基本语境条件，在于智能体欲获得某些话题C的更多信息。在此主要的目标，一般为获取知识和新奇感；而取悦谈话对象为其次。
2. 另一种语境情况，为智能体知道谈话对象欲谈论话题C，因此智能体决定针对C发问、或提出与C相关的问题。若是这样的情况，取悦谈话对象会是主要目标，而获取知识或新奇感则是其次。

* 程序：语意内容为”你对C有何看法？”(”What do you think about C?”)、”C的真伪值为何？”(”What is the truth value of C?”)、或”关于C你知道些什么？”，但这些疑问会以惯用语的方式表达。
* 附加问句
* 例如：”对吧？”(”Right?”)
* 相关目标、语境：这部分的关键语境条件，在于智能体欲对谈话对象发问，而智能体几乎肯定该问题的正确答案–但智能体欲确定谈话对象也认同。在此主要的目标，一般为获取知识和新奇感；而取悦谈话对象为其次。
* 程序：此程序为通过确切阐述为连续的问题（”X，那你同意X吗？””X.Do you agree with X?”），提出”你同意X吗？”(”Do you agree with X?”)的问题。以惯用语表达而言，在连续问题中的第二个部分，“X“通常会被去除，措辞表达的方式有很多种。
* 陈述性特殊疑问句(Declarative Wh-question)
* 例如：”你跟他们说了什么？”(”You told them what?”)
* 相关目标、语境：在此的关键目标，一般为获取知识和新奇感；而取悦谈话对象为其次。引起此等阐述的关键语境，似乎是在VariableNode的重要性比问句中的其他字词还要高的情况；因为这类阐述更强调了疑问词。
* 程序：这部分的语意内容纯粹只是个疑问句；必须指示微规划系统以陈述性的问句形式来表达疑问。
* 选择式问句[[5]](#footnote-6)
* 例如：”你认为谁最有可能当选：希拉里、杰布•布什还是迈提•毛斯”(”Who do you think is more likely to get elected：Hillary,Jeb Bush or Mighty Mouse?”)
* 相关目标、语境：在此的关键目标为获取知识和新奇感；取悦谈话对象为其次。关键语境在于智能体欲从谈话对象获得某问题的答案，而智能体认为该问题仅有相当少数的可能答案。另一种语境，在于智能体想要的问题答案是以反意的形式（OrLink或XORLink）出现在Atomspace中。
* 程序：语意内容出于”问题：候选回答1，……，或候选回答K。”(”Question：Candidate Answer 1,...,or Candidate Answer k.”)之形式。这般言辞可直接或以惯用语方式传达。

### 7. 陈述

* 非意见式陈述
* 例如：”猫不冬眠。”(”Cats don’t hibernate.”)
* 相关目标、语境：这部分的关键语境有二：

1. 谈话对象对智能体问了一个非真伪值的疑问句，而智能体欲给予回答。在此情况下，取悦谈话对象的目标相当强烈。
2. 智能体在其AttentionalFocus中有些具高度重要性的Atom，并想明确地表达出。在此情况下，取悦谈话对象的目标较不那么强烈。若智能体欲获取涉及Atom的一般知识，这时对获取知识的目标会有强烈的连结；而如果智能体发现新奇感往往是通过发问这些Atom、或相似Atom而产生，对新奇感的目标则有强烈的连结。

* –程序：将问题中的Atom传送至微规划系统进行言语表达即可。
* 意见式陈述
* 例如：”我不认为希拉里•克林顿会当选总统。”(”I don’t think Hillary Clinton will be elected President.”)
* 相关目标、语境：除了牵涉的Atom具有较低的置信度，或智能体怀疑谈话对象强烈不认同智能体的真伪值评估之情况，其他情形与”非意见式陈述”相同。
* 程序：语意内容为”我认为X”(“I think X”)或”我不认为X”(“I don’t think X”)。这般言辞可直接或以惯用语方式传达。

### 自我描述

* 提议、选择、承诺
* 例如：”我会考虑的。”(”I’ll think about it”)
* 相关目标、语境：谈话对象要求智能体做某事。智能体必须决定自己是否愿意做这件事。这部分一般目的是取悦谈话对象。
* 程序：语意内容为”我会试着X”(”I will try to do X”)（若智能体不确定是否会成功）、”我会X”(”I will do X”)（若相当肯定会成功）、”我不想X”(”I don’t want to do X”)、”我会试着X，但我不确定是否能成功”(”I will try do to X,but am not sure I can succeed”)（若智能体认为成功的机会很低）、”我会想想我是否能够X”(”I will think about whether I can do X”)、或”我会思考我是否想要X”(”I will think about whether I want to do X”)。所有这些言辞都可直接或以惯用语方式表达。
* 自言自语
* 例如：”嗯……我想知道……”(”Hmmm,well I wonder....”)
* 相关目标、语境：一组Atom在智能体的AttentionalFocus中产生高度的重要性，并且与当前对话中产生之Atom有合理、紧密的关联性。这些Atom可代表某陈述句或问句（后者的情况为这组Atom具有虚悬不定、无约束的VariableNode；或者其不具有VariableNode、但置信度低的情况）。这里的主要目标为取悦谈话对象和求知。
* 程序：语意内容为”我在思考X”(”I am thinking X”)（针对某陈述）、”我很好奇是否X”(”I am wondering if X”)（针对置信度低的陈述X）、”我在想X”(”I am wondering X”)（针对某非真伪值的疑问，例如某X带有虚悬不定的VariableNode）。所有这些言辞都可直接或以惯用语方式表达。

### 后设话语Meta-Utterances

* 模棱话
* 例如：“我不太确定我的背景经历是否足以回答这问题，但……”(“I’m not so sure I have the background to really answer that,but...”)
* 相关目标、语境：在此的语境条件，在于智能体欲做出陈述，但对该语句仅持有较低的置信度。
* 程序：语意内容为“我对接下来要将的话较没有把握，但：S”(”I have relatively low con-ﬁdence in the following statement,but：S”)、或”我对接下来要将的话约有三分的把握，但：S”(”My conﬁdence in the following statement is roughly.3,but：S”)。所有这些言辞都可直接或以惯用语方式表达。
* 答复、同意前的保留
* 例如：”等等……让我想一下”(”Wait...give me a minute....”)
* 相关目标、语境：在此的语境条件，在于智能体正为接下来所要说的话做内部处理，而这个程序耗费的时间超出了某”可接受的对话迟滞”参数值（可根据某特定谈话语句之间的平均迟滞时间、或与相同谈话对象其他对谈语句之间的平均迟滞时间、或者其他类似方式的谈话等情况设置之默认值改编）。
* 程序：
* 不理解的信号
* 例如：”你在说什么鬼话呀？讨厌的家伙！”(”What in frick’s sake are you talking about,pesky human??!”)
* 相关目标、语境：在此的主要目标为获取知识，取悦谈话对象的动机较弱。语境条件在于谈话对象说了某些智能体认为荒谬的话。另一较不重要的语境条件，在于谈话对象说了某些对智能体而言错误至极的话。
* 程序：语意内容为”我不明白你说的X”(”I don’t understand what you mean by X”)，所有这些言辞都可直接或以惯用语方式表达。

### 10. 命令与建议

* 动作指令
* 例如：“告诉我他说了什么！”(“Tell me what he said!”)
* 相关目标、语境：在单纯的对话系统语境中，切题的语境条件在于智能体欲获得某些其高度重视的信息，而这些信息是智能体认为谈话对象所具备的。另一相关语境，在于智能体已向谈话对象询问某特定问题，然而谈话对象的答复似乎不是该问题的正确解答。在某个应用程序中，CogDial被搭配一个能够执行非言语行为之成体运作，该程序中也有这样的语境；智能体希望谈话对象执行某个动作（因谈话对象做了这个动作会使智能体更完善达成目标），而相信智能体有能力做这个动作，是有原因的。
* 程序：语意内容为“我要你告诉我X”(“I want you to tell me X”)或“我希望你X”(“I want you to do X”)，这般言辞可明确或以惯用语方式表达。
* 第三人称谈话
* 例如：”说真的，鲍勃，你是这么认为的吗？”(”Really,Bob,do you feel that way?”)
* 相关目标、语境：此处的相关目标为取悦谈话对象，完全在于情感上的关系管理。导致这类话语的语境条件有许多种，其中几种可能为：

1. 谈话对象回答了某问题，其答案看似与多数人、或多数与谈话对象普遍相似的人回答的答案有所分歧。
2. 智能体问谈话对象一个问题，智能体怀疑谈话对象的回答会与多数人、或大部分与谈话对象普遍相似的人会回答的答案有所分歧。
3. 问的问题被判断为”私人”，例如一般亲密的友人或家人才会问的问题。
4. 做出了有关谈话对象的陈述，就谈话对象的观点来说，是”私人”的陈述。

* 程序：最好将此视为可穿插到多种其他言语行为之言语现象，而非个别的言语行为。在另一言语行为制造出一组待传入微规划系统的Atom后，将会调用”可能插入第三方谈话”的规划器，此采用的准则包含前述的相关语境，根据这些准则，决定是否将谈话对象的名字引入到微规划系统被予以处理的一组Atom中（将名字给予此组Atom，会导致表面生成器最终产生出”第三人称谈话”的例句）。

## 本章小结

XXX介绍本文本文实现的基于上述模型的智能对话系统的框架，并引出下面章节中的自然语言理解和自然语言生成系统。

第五章 自然语言理解：从语言到逻辑

本章阐述如何将自然语言语句转换成第二章中介绍的基于超图的抽象语义逻辑形式。我们认为，借助于依存关系语法和传统逻辑与谓词逻辑的合理结合，将自然语言表达式转换成满足下列两个条件的逻辑表达方式是完全可行的：

* 包含该自然语言表达式的主要语义
* 具体化自然语言表达式中存在的任何无法在语言到逻辑的转换消除的歧义，使得这些歧义能通过基于语境知识的逻辑推理后很直截了当地得到消除。

鉴于此，我们采用的自然语言理解的方法是，首先利用链语法分析工具Link Parser[109]，然后在此基础上搭建了一个用于依存关系抽取的工具RelEx，最后开发了一个新的逻辑关系抽取工具RelEx2Logic，通过超图的同态映射方式将句法结构转换成语义表示。该方法概念上的本质并不依赖这些特定的工具。

确切地说，如何将“自然语言理解”分解成不同模块以及如何进行不同模块之间的转换，取决于对语言学理论的选择。在2008-2012年期间，我们的自然语言理解模块采用如下流程：

文本 --> 分词/断句 --> 基于链语法的句法分析（Link Parser） --> 依存关系抽取（RelEx）--> 基 于 FrameNet 的 语 义 关 系 抽 取（RelEx2Frame） --> 语 义 节 点 和 关 系 链（SemanticNodes & Links）

2012年的时候我们对系统进行了一些简化，取消了对FrameNet的依赖，采用了如下的目前正在使用的系统：

文本 --> 分词/断句 --> 基于链语法的句法分析（Link Parser） --> 依存关系抽取（RelEx）--> 抽象的逻辑关系抽取（RelEx2Logic --> 语义节点和关系链（SemanticNodes & Links）

需要注意的是，目前的很多自然语言理解系统都有“词性标注”阶段，在我们目前使用的方法中，词性标注被绑定在句法分析阶段，对于Link Parser来说，词性已被作为单词的一个属性定义在词典里。尽管如此，如果能在Link Parser里使用先进词性标注技术，无疑能减少不少字典编写方面的工作，也能在一定程度上指导句法分析过程，因此也是有一定存在意义的。

本文的工作已经表明，上述系统中的各模块操作都是可行的，只需针对每一步制定相应的规则，或者通过有监督的机器学习方法来学习相应的规则，来指导其中的操作。针对本文自然语言理解方面的以下几个额外事项将在未来的研究中被实现：

* 使得各个子流程中所使用的规则，能很自然地支持基于持续经验增长的修正和泛化。
* 使得语义理解能根据特定的语境来指导规则的选择。
* 知道何时该打破规则，而根据语义的直觉指导相关操作。

另外需要注意的一点是，当使用基于规则的方法时，在规则的设计需要特别注意规则的可扩展性和可普及性，因为随着系统的经验增长，原先设计的规则可能无法被满足。

本章接下来的章节会系统介绍我们使用的自然语言理解系统中的各个子系统的工作原理和方法。

## 链语法

本小节主要介绍链语法的基本原理和方法，并分析了和语言形式主义的不同，同时指出其中的问题以及我们对其进行的一些改进。

链语法（Link Grammar）在1991年由Daniel Sleator和Davy Temperley共同提出[]。它和被广泛应用的依存语法类似，但两者也有很大的不同，比如链语法中两个单词的连接是无序的，而依存语法中有依存和被依存关系。链语法允许句子中的链接有环状结构，而依存语法是不允许环的存在的。链语法更倾向于语法理伦的词汇主义。

链语法的核心是链语法词典（Link Dictionary），词典中的每一个词都记录着一些特点的链接要求，这些链接要求通过一系列具有特定逻辑排序的链接子（connector）组合成的公式（Formula）收录在词典的相应词条里，链接子包含名称和后缀（“+”或“-”），后缀表示该链接子的指向方向,“+”表示该链接子指向右，“-”表示该链接子。如果两个词需要合法连接，则要求两个词首先有相同名称的链接子，且左边单词的链接子的连接方向必须指向右，右边单词的链接子的连接方向必须指向左。

链语法分析器（Link Parser）是基于链语法的句法分析器。使用链语法分析句子的时候，句法分析器对句子中的每个单词去查询链语法词典得到每个单词的链接要求，根据这些链接要求进行相应的链运算，便得到句子的链语法结构。链语法判断句子是否合法，除了需要满足句子所有单词的链要求外，还要满足以下四个原则：

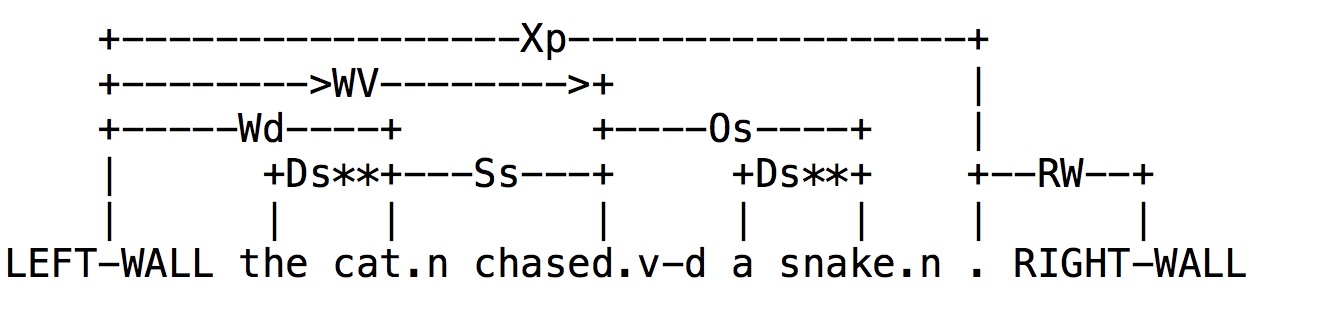
ss平面性(Planarity)，链之间不能互相交叉。

* 连通性(Connectivity)，句子中的每个单词都必须有链和其他至少一个单词相连，形成连通图。
* 顺序性(Ordering)，公式中较左边的链接子必须和距离单词较近的单词链接，反之，公式中较右边的链接子必须和距离单词较远的单词链接。
* 排它性(Exclusion)，一对单词之间同时不能有两条链链接。

下面我们借用Sleator和Temperley的“Parsing with a Link Grammar”[]中使用的例句来解释链语法分析的操作过程。

**The cat chased a snake**

链语法分析器对上面的例子分析后产生如下结果：



由于我们对该链语法分析器做了不少的改进，所有我们这里给出的分析结果和Sleator和Temperley的原文中的结果以及发布在网上的分析器得到的结果会有细微的不同，后面章节会介绍我们做的改进。

链语法认为在句子最前面加上一个虚拟词（通常用LEFT-WALL表示）是很有必要的，这样首先能保证链语法的“连通性“原则，使得句子最后的标点符号能与LEFT-WALL连接而不被孤立。另外，在本文的5.1.2节也提到，通过这样的虚拟词来追溯句子的头（主动词）也是非常方便可行的。上面的句法分析结果中出现的”RIGHT-WALL”是可选的，可以用于特殊标点符号的处理，但通常情况下只是用RW链与句子最后的标点符号相连。

前面提到链语法词典是链语法的核心，它包含所有常用的英文单词的链接要求。下方的表格列出上述例句中出现的单词以及它们在链语法词典里被定义的链接要求。

表5.1

|  |  |
| --- | --- |
| 单词 | 公式 |
| a, the | D+ |
| snake, cat | D- & (O- or S+) |
| Chased | S- & O+ |

如前面所述，链接要求限制了这些连接子必须按照一定的原则来分配，例如“the”只有一个向右的连接子D+，那么它只能和带有D-的单词形成合法的链接。而“snake”和“cat”都能和它相连，但根据链语法的“顺序性原则”，“cat”比“snake”近，所以“the”“cat”之间可以画一条D链，同样的道理，“chased”和“snake”之间可以画一条O链，以此类推。最终我们可以得到如下的结构图。需要注意的是，我们这里只是对链语法的工作原理做简单地介绍，因此忽略了链接子类型的子类型等细节，比如“cat”含有Ss+，“chased”含有链接子Ss-,，所以该句法结构图和我们上面列出的目前链语法分析器在连接类型上有点小出入。

链语法分析器在对句子进行句法分析时，对句子中的每个单词，都会考虑下面两种变量：

* 该单词在链语法词典中对应的链接要求，即上面中的公式（Formula）
* 该单词为了满足句子结构的一致性（Agreement）必须具备的特征属性

比如在上述例句中，对于单词“snake”，通过查询链语法词典，我们得到相应的公式“D-&(O-or S+)”，同时该单词还有相应的特征属性如“时态（tense）”“人称（person）”等。但对于单词“the”，就不需要与一致性相关的变量。

链语法分析器中也使用简单的转换生成类似短语结构语法的句法分析结果，如下：

(S (NP The cat)

(VP chased

(NP a snake))

.)

在我们的工作中，对该短语结构用的不多，所以这里不做详细说明。我们会在下一节简单讨论链语法和短语结构语法的潜在关系。

### 5.1.1 链语法与短语结构语法

讨论各种语法的优缺点不是本文的重点，本节只将链语法和典型的短语结构语法做个简单的对比来讨论它们的潜在联系。一般来说，依存语法和短语句法也有相应的联系（参见图5.15.1），但不同的依存语法使用不同依存关系集合，也有不同的属性，分析起来会比较复杂，也不是本文的研究内容，所以我们这里不做详细阐述。

简单起见，这里只列出两条有用的观察结果，基于这些观察结果，我们不难发现，在链语法中也隐形存在短语结构。这是有一定道理的，但由于自然语言的复杂性，某些情况下可能也不是那么简单。

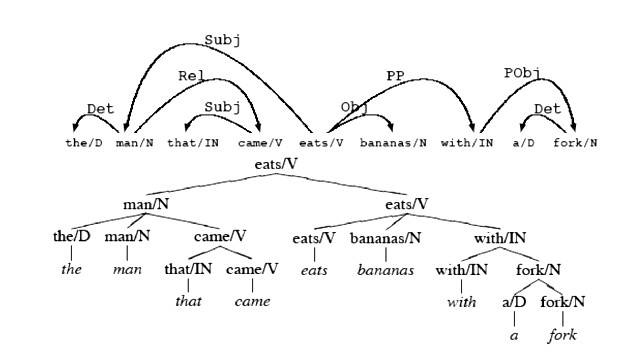


图 5.1 依存句法分析和短语句法分析对比[[6]](#footnote-7)

* 链语法中的公式是符合语法范畴的。例如，前面提到的例句中“chased”的链结构是“S-&O+”，在范畴语法中，这意味着，“chased”属于这样的语法范畴，该范畴中的词都满足链结构“S-&O+”。换句话说，链语法中每一个“公式”都对应依附该公式的单词的范畴。
* 词之间的链接也可以看成是短语的核心词（head）之间的链接。例如，在例句“The cat chased a snake”中，“chased”和“snake”之间有链接O，那么也可以说，以“chased”为核心词的短语和以“snake”为核心词的短语之间有链接O。换句话说，可以理解成，链语法为了简化，通过核心词来识别短语。

### 5.1.2 识别句子的中心词

链语法从词的局部着手关注任意两个词之间的关系，在一定程度上忽略了语言的层次结构。因此标准的链语法也存一些弊端，有不少语言现象会被链语法拒绝认为是不合法的结构。比如并列结构、介词短语等。在本文的研究过程，我们对链语法做了很多的改进，由于链语法的链接要求非常复杂，修改链语法词典可能牵扯到很多语言学上的问题。但这不是本文的研究重点，所以这里不一一列出我们做过的改进，只简单的列出几项较大的改进：

* 改进了链语法对连词的处理
* 改进了链语法对量词的处理
* 识别句子的中心词。

链语法分析器（Link Parser）是一个开源工具，包括链语法词典，我们所有的改进都发布在[http：//abisource.com/projects/link-grammar/](http://abisource.com/projects/link-grammar/).

为了读者对如何修改链语法词典有个大致的思路，本小节就其中的改进“识别句子的中心词”展开讨论。

对于“识别句子的中心词”的改进，我们的出发点是使链语法分析结构（Link Parse）能转换成一颗类似的依存句法树。改进后，我们能直接通过追溯“WV”链找到句子的中心词，或者通过追溯“CV”链找到子句的中心词。找到中心词后，可以将其定为树的根节点，然后依次遍历各个链，最终能得到一颗句法树。具有这样的转换能力有以下优点：

* 能使用很多能用在树结构上的机器学习方法来改进链语法分析器，比如频繁子树挖掘(Frequent Subtree Mining)[110]
* 能更直接的链语法分析结果转换成依存句法树，从而能使用依存语法的语料库或者工具来改进链语法分析器
* 能更好地从其他语法理论（例如词语法（Word Grammar）[111]）角度来解析链语法分析结构
* 能简化一些RelEx中需要通过追溯很多链来找到中心词的规则

对于句子的中心词定义，语言学界有很多不同的看法。我们这里借鉴依存语法的观点，选择使用句子的主动词来作为句子的中心词，以及子句的主动词来作为子句的中心词。

此项改进的具体目标就是使句子的中心词，即句子的主动词，更容易从链语法分析得到的结果中被检索到。这样的改进不仅仅使链语法分析结构能够很直接地转换成类似依存句法树，因为依存语法一般以主动词为根节点；还无意中改进了链语法分析器对复句的分析能力，使其产生了更直观合理的分析结果，因为我们采用了将功能词与子句的主动词相连的方法来追溯子句的中心词，这显然比标准链语法中，不论什么情况，都使用功能词与子句的第一个词互相链接更直观合理；这样的改进给我们后续研究的依存关系抽取和逻辑关系抽取带来了诸多方便，因为在改进之前，无聊是并列复句还是偏正复句，或者是带状语的单句，链语法都将其中功能词与子句的第一个词相连，这样使得判断子句之间的语义逻辑关系变得很困难。

为了能实现这样的改进，我们首先引入了三种标准链语法中不存在的链接类型：WV，CV和IV。其中，WV用于链接LEFT-WALL和句子的中心词，CV用于链接功能词和子句的中心词，IV用于链接LEFT-WALL和非限定动词。这些链接类型并不是第一批用来处理句子中心词的链接类型，链语法词典中有几个链接类型已经充当类似的角色，如B，AF，CP,Eq，COq等，但是这些链类型的用法非常复杂而且分类很模糊，引入的这几个新的链接类型，使句子的中心词角色更突出和直观，同时也和其他语法接轨。有关不同的链接语法中的链接类型的含义和使用方法，可参考链语法词典的在线文档：[http：//www.abisource.org/projects/link-grammar/dict/section-WV.html](http://www.abisource.org/projects/link-grammar/dict/section-WV.html)

对这一改进，我们采用的具体实现方法可以归纳为以下几步：

1. 修改并增强链接B，使得B能指向句子的中心词，以及子句的中心词。上面一段提到，链语法词典中的链接类型B有类似能指向句子的主动词的功能，但是其使用方法和能被应用的情况分类很模糊。
2. 通过链接B找到句子的中心词，然后将LEFT-WALL和该中心词之间用新引入的W链接相连。
3. 通过B链接找到子句的中心词，修改链接CO或者Cs（CO和Cs都是改进前链语法中用于链接功能词和子句的第一个词的链接类型），使其指向子句的中心词，并将其链接类型改为CV。

IV 的处理方式和 WV 类似，这里不再详述。

下面给出了针对这些关系链接的改进的例子，例句：“Call me when you are ready.” 改进前句法分析结果如下：

+------------------Xp-------------------+

| +---MVs--+ |

+---Wi---+-Ox-+ +-Cs+-Spx+--Pa--+ |

| | | | | | | | LEFT-WALL call.v me when you are.v ready.a .

改进后句法分析结果如下：

+------------------Xp-------------------+

| +---MVs--+---CV---+ |

+---Wi---+-Ox-+ + +-Spx+--Pa--+ |

| | | | | | | | LEFT-WALL call.v me when you are.v ready.a .

例句：”I left soon after I saw you.” 改进前句法分析结果如下：

+--------------------Xp---------------------+

| +----MVs----+ |

+--Wd--+-Sp\*i-+--MVa-+ +-Cs+Sp\*i+-Ox-+ |

| | | | | | | | | LEFT-WALL I.p left.v-d soon when I.p saw.w you .

改进后句法分析结果如下：

+--------------------Xp---------------------+

+------WV-----+----MVs----+---CV---+ |

+--Wd--+-Sp\*i-+--MVa-+ + +Sp\*i+-Ox-+ |

| | | | | | | | | LEFT-WALL I.p left.v-d soon when I.p saw.w you .

例句：”Apparently, she loves cheese.” 改进前句法分析结果如下：

+--------------------Xp--------------------+

+---------Wd--------+ |

| +---CO---+ |

| +--Xc-+ +--Ss-+----Ou---+ |

| | | | | | | LEFT-WALL apparently , she loves.v cheese.n-u .

改进后句法分析结果如下：

+--------------------Xp--------------------+

+------------WV-----------+ |

+---------Wd--------+ | |

| +---CO---+ | |

| +--Xc-+ +--Ss-+----Ou---+ |

| | | | | | | LEFT-WALL apparently , she loves.v cheese.n-u .

需要说明的是，在链语法分析器的当前版本中，虽然我们采用了WV链来链接LEFT-WALL和句子的中心词，但我们仍然保留了原来版本中的链接LEFT-WALL和句子的第一个词的Wd或者Wi等链，只是为了保持版本的向后兼容性。在转换成句法分析树的时候，我们会忽略Wd或者Wi等这些链。

### RelEx

这一节我们将介绍依存关系抽取工具RelEx（Relation Extractor）的工作原理和基本实现方法。RelEx采用上一节中链语法分析器的输出作为输入，将其转换成一个特殊的特征结构图，然后根据相应的规则进行一系列的图结构转换，最后得到一个含有比链语法分析结果更抽象一点的介于句法关系和语义关系之间（syntactico-semantic）的依存关系图。

RelEx包含了很多个模块，我们这里只介绍其中关键的模块。它的核心思想就是将链语法分析器的分析结果转换成一个特有的特征结构有向图（Feature Structure），然后使用一系列规则（在RelEx系统中被称为句子算法，Sentence Algorithm）对特征结构图进行一系列相应的有序的修改，最终得到一个精炼的特征结构图。最终得到的特征结构图中包含了句子中词和词之间的RelEx句法语义关系（如主谓关系、动宾关系等），还包含了句子中每个词的相关属性（如词性、时态等）。RelEx还处理一些消歧工作，如指代消解，我们会在下面章节进一步阐述。

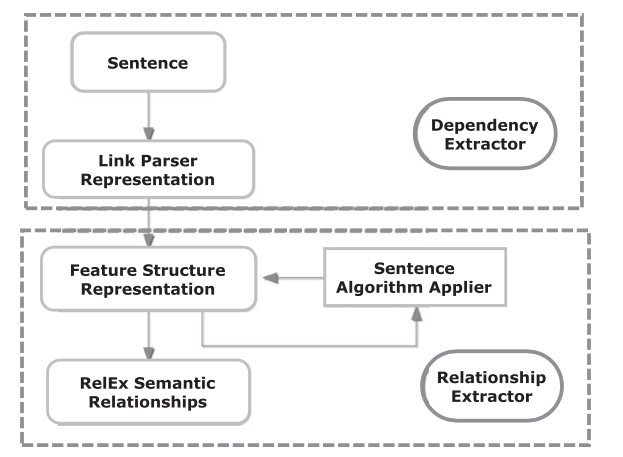


图 5.2 RelEx框架图

RelEx的输出可看作是被简化了的输入句子的语法结构，在一定程度上，RelEx还对一些表层关系进行归一化处理，因为很多等效但异构的动词框架会被映射到一致或者同态的特征结构图中。例如下面的两个不同的句子语义却一样：

Mary ate the cake.

The cake was eaten by Mary.

在RelEx的输出中，它们同时含有下面的RelEx依存关系：

\_subj(ate, Mary)

\_obj(ate, cake)

### RelEx 的系统框架

RelEx系统框架主要包括两大模块：附属关系抽取模块和语法关系抽取（参见图5.2）。它能识别句子中词和词之间的主语、宾语、间接宾语和其他依存关系，也能像其他依存语法分析器那样生成依存关系树。

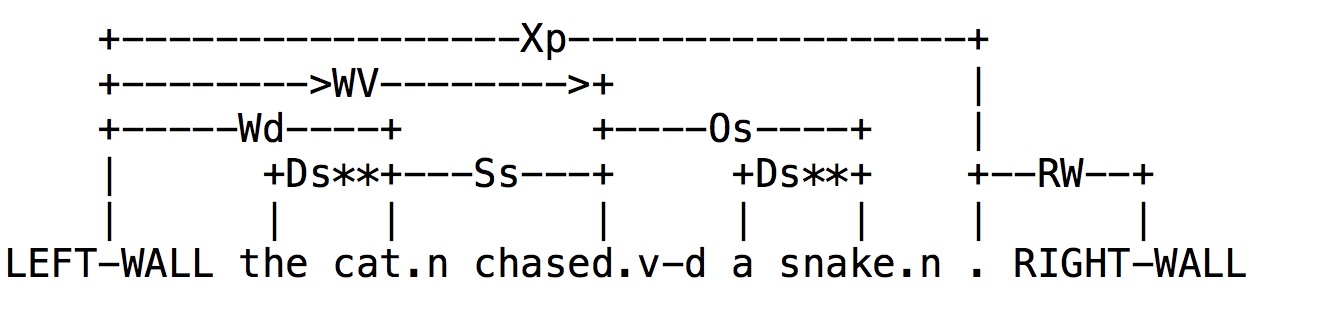
RelEx首先对句子中的每个词创建一个特征节点（FeatureNode），然后通过一系列相关规则不断更新每个特征节点。这些规则将链语法分析器的输出结果中的不同链的组合转换成相应的RelEx依存关系，有些转换可能是通过某一条规则直接从链到RelEx依存关系的转换，也可能是间接地根据几条规则动态修改一个词的特征节点，并结合其他相关的特征节点中的特征，最终得到相应的RelEx关系。

接下来我们使用在链语法章节中使用的例句“The cat chased a snake.”来更详细地解析RelEx的每个步骤和实现方法。

**步骤1：将链语法分析器的输出转换成一个特征结构图。**

如上所述，RelEx首先将链语法分析器的输出结果转换成一个特征结构图。特征结构图是一个带权有向图，其中节点可以表示一个值，也可以是一个无序的特征列表。RelEx中，特征指的就是一条指向另一个节点的带权边，而特征的值通常是一个字符串。

从上一节中我们可知，例句输入链语法分析器后产生下面的链语法结构：



根据我们上面的步骤描述，首先需要将上面的链语法分析结构转换成特定的特征结构图。句子中的每个词都会被转换成特征结构图的一个节点，其中该节点包含以下特征（[]中的内容表示特征的类型，如NEXT[node]表示该特征是一个特征节点类型）：

* NEXT[node]：该特征指向表示该词的后一个词（如果存在）的特征节点。
* PREV[node]：该特征指向表示该词的前一个词（如果存在）的特征节点。
* this[node]：该特征指向自己（该特征的存在只是为了方便某些句子算法的执行）
* wall[node]：该特征指向表示 LEFT-WALL 的特征节点。
* index\_in\_sentence [int]： 该词在句子的位置，LEFT-WALL 的位置为 0，以此类推。
* start\_char [int]： 该词的第一个字符在原句子中的位置。从 0 开始计算。
* end\_char [int]： 该词的最后一个字符的后一个字符在原句中的位置。end\_char - start\_char = 字长
* str [string]： 该词的内容。该内容应该和从句子中取（start\_char, end\_char）之 间的子串得到的结果一致。
* POS [string]： 该词的词性。
* num\_left\_links [int]： 从该词出发指向左方向的链接个数。
* num\_right\_links [int]： 从该词出发指向右方向的链接个数。
* linkL0 [node]： 指向表示从该词出发指向左方向的链接的特征节点
* linkL(1,2,3,...) [node]： 指向表示从该词出发指向左方向的链接的特征节点（如果从该词出发指向左方向的链接个数超过1）
* linkR0 [node]： 指向表示从该词出发指向右方向的链接的特征节点
* linkR(1,2,3,...) [node]： 指向表示从该词出发指向右方向的链接的特征节点（如果从该词出发指向左方向的链接个数超过1）

类似地，链语法分析结构中的每个链接也被表示成一个特征节点，其中该节点包含以下特征：

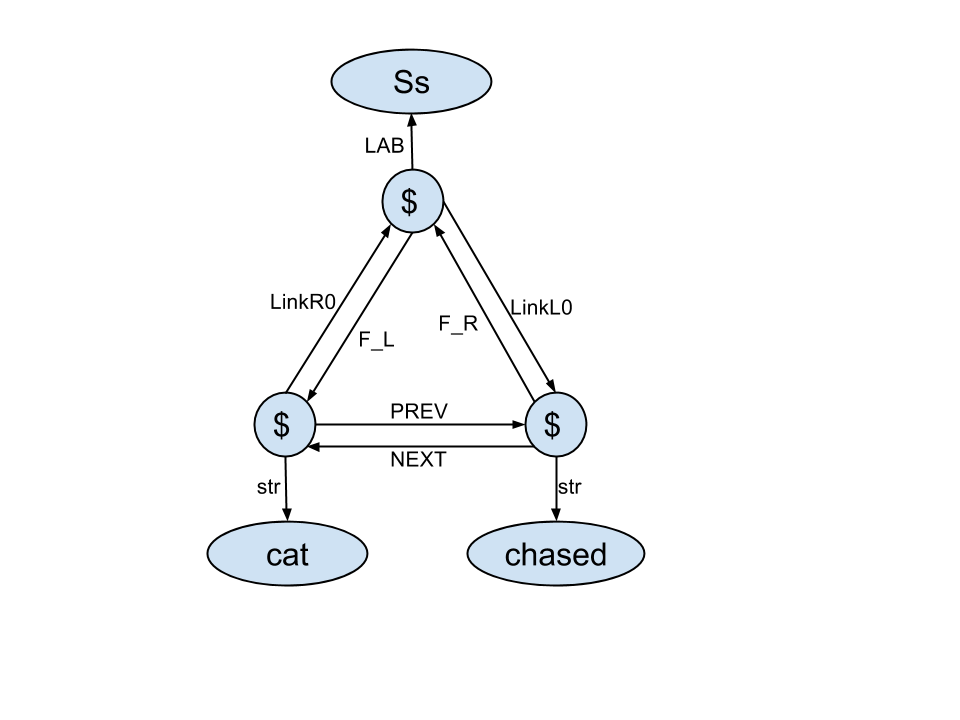


图 5.3 初始特征结构图

* LAB [string]： 该链接的名称
* F\_L[node]： 指向表示该链接左边的词的节点
* F\_R[node]： 指向表示该链接右边的词的节点

不难发现，RelEx中的使用特征结构图非常复杂，为了能更形象地理解特征结构图，本文截取了上述例句的特征结构图的部分来进一步解释（参见图5.35.3）。

图5.35.3中，带$的节点可以表示代表句子中的某个词的特征节点，通过特征str指向词的内容；也可以表示代表某个链接的特征节点，通过特征LAB指向链接名称。从上图我们可以看出，“cat”和“snake”通过链接Ss相连。例句中的第二个个词节点的特征str表示该词的内容“cat”，特征NEXT指向表示下一个词的特征节点（即str为“chased”的节点），特征F\_L表示该词是链接“Ss”的左右的词。“LinkR0”表示它指向的链接Ss是从该词“cat”出发指向右边的链接。以此类推，不难从上图中找出例句中第三个词节点的一系列相关特征。

**步骤2：执行包含一系列句子算法的句子算法应用器（Sentence AlgorithmApplier），对步骤1中得到的特征结构图进行相应的改写操作。**

句子算法应用器从句子算法的定义文件中导入一系列的句子算法，句子算法的执行顺序必须按照定义文件中句子算法的排序。当一个句子算法被执行时，RelEx不断迭代遍历特征结构图中的每一个特征节点，对每个节点都尝试执行该句子算法。句子算法的执行会导致特征结构图的修改，因此它们的操作顺序是很重要的。例如，假如前面的句子算法执行后，特征结构中的所有特征节点的词性（POS）特征都被删除，那么用于处理含有POS特征节点的句子算法将无法被执行。

所有的句子算法都带有一个特定的标记（可以理解成该句子算法的名称），当某个句子算法被成功执行后，该算法会在被成功执行该算法的节点上添加一个特征SIG，并将其赋值为该句子算法名称。被句子算法添加的一个最重要的特征是指向表示词的特征节点的ref，RelEx使用ref特征来输出最后的关系集合。

除了复杂的特征结构，RelEx的另一个核心部分是前文一直提到的句子算法。RelEx使用的句子算法也是很复杂的，可分为抽象的句子算法和需要实例化的句子算法。我们这里选择一个用的最广泛的通用句子算法TemplateActionAlg来进一步阐述。一个TemplateActionAlg实例能接受文本输入，第一行表示该实例的名称（不可重复），接下来是一个用于匹配的模板路径集合（Template Paths和行为路径集合（Action Paths），之间用“=”隔开。对于模板路径<x,y,z>，表示目标值按照<x,y,z>这样的路径来解析。对于操作路径<x,y,z>，表示目标值按照<x,y,z>这样的路径被赋值。例如，在如下的被实例化的TemplateActionAlg的句子算法中，

#TemplateActionAlg SPECIAL-ADJ

; Used for "easy to read."

; The B links back to the adjective. But it should really be

; interpretted as linking back to the subject of the adjective

<LAB> = \B\.\*|\BW\.\*

<F\_L POS> = adj

=

<F\_R obj> = %

<F\_R obj> = <F\_L subj>

<F\_L ADJ-OBJ-FLAG> = T

在该句子算法中，模板就是：

<LAB> = \B\.\*|\BW\.\*

<F\_L POS> = adj

不难看出，第一行表示用于匹配那些表示链接名称B或者BW的特征节点，句子算法的表示使用了正则表达式，意味着，只要链接名称以B或者BW的特征节点都满足。除此之外，要想上面的句子算法能被执行，还要满足第二行，即该特征节点表示的链接左边的词必须是一个形容词。

如果上面的两条模板都匹配成功，那么下面的操作将会被执行：

<F\_R obj> = %

<F\_R obj> = <F\_L subj>

<F\_L ADJ-OBJ-FLAG> = T

这些操作可被解释如下：第一行的操作表示该链接的右边的词所在的特征节点的obj特征会被清空。清空后，紧接着将该链接左边的词所在的特征节点的sub特征值赋给它。最后，将该链接的左边的词所在的特征节点的ADJ-OBJ-FLAG特征值设为T（真）。

我们接着用上面的例句“The cat chased a snake.”，接下来两个图是图的节选特征结构图，更直观地解释了，通过执行相关的句子算法后，被修改的结构特征图。

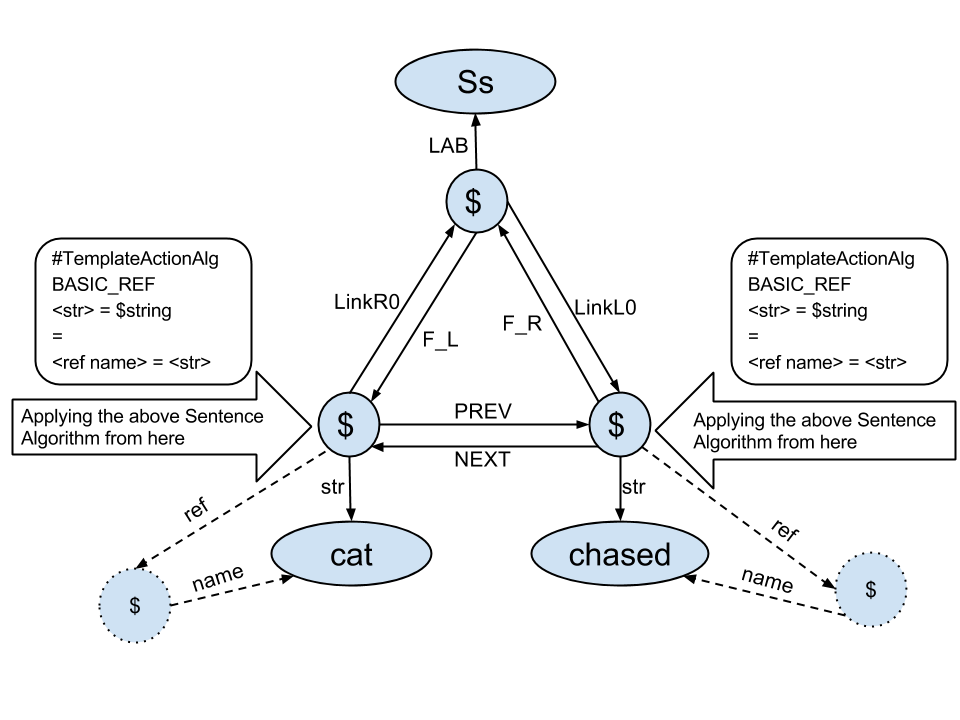


图 5.4 修改过的特征结构图

图5.45.4中虚线部分是执行了下面的句子算法后特征结构图被修改的部分：

#TemplateActionAlg

BASIC\_REF

<str> = $string

=

<ref name> = <str>

根据上面的句子算法，当模板“<str> = $string”被匹配时，会执行操作“<ref name>=<str>”，也就是，把该词的内容赋值给通过路径<ref, name>找到的节点。

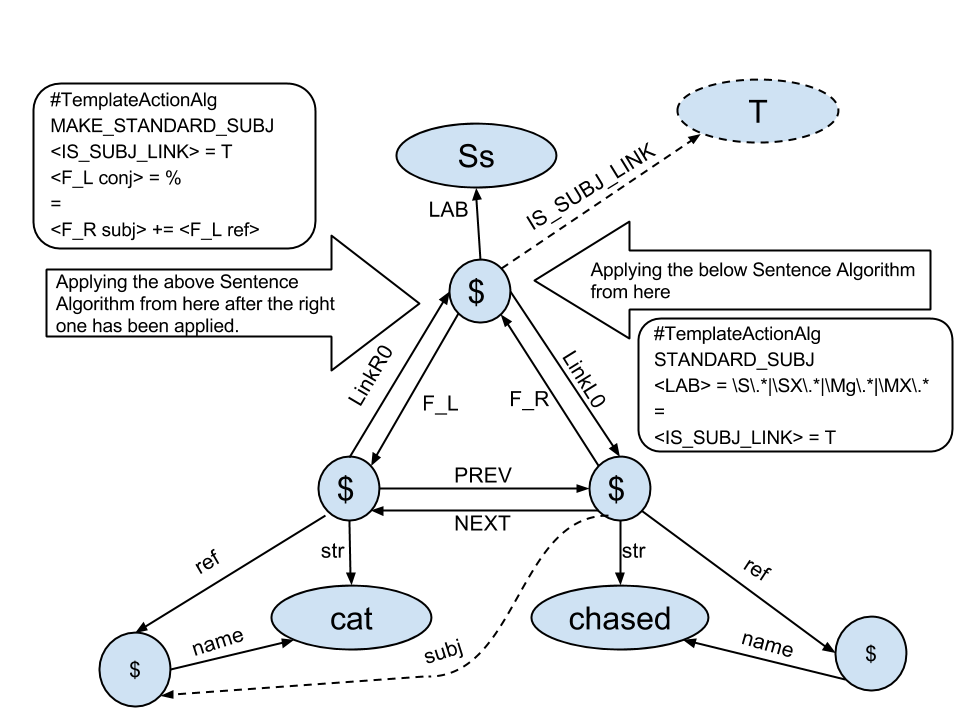


图 5.5 修改过的特征结构图

图[5.5](#_bookmark76)中，同样地，虚线部分表示执行相应的句子算法后特征结构图被修改的 部分。其中使用的句子算法有：

#TemplateActionAlg

STANDARD\_SUBJ

<LAB> = \S\.\*|\SX\.\*|\Mg\.\*|\MX\.\*

<LAB> != \MX.[rj]\.\*

<LAB> != MX|MXs|MXp

=

<IS\_SUBJ\_LINK> = T

如果找到连接主语的链接，那么创建一个名为 sub 的依存关系：

#TemplateActionAlg

MAKE\_STANDARD\_SUBJ

<IS\_SUBJ\_LINK> = T

<F\_L conj> = %

=

<F\_R subj> += <F\_L ref>

第一个句子算法中表明，当找到一个特征节点的“LAB”特征满足这样的正则表达式\S\.\*|\SX\.\*|\Mg\.\*|\MX\.\*时，即图5.5中最上面的特征节点的情况，那么会创建一个新的特征IS\_SUBJ\_LINK并赋值为”T”（真）。IS\_SUBJ\_LINK是一个临时特征，会在上面第二个句子算法中用到，类似地，在执行了第一个句子算法后，图中最上面的节点也满足了第二个句子算法的条件，于是会接着执行第二个句子算法，即执行<FR sub j>+=<FLre f>，这说明，系统会将FLre f的值传到通过路径FR sub j的相应特征上，也就是上图中最下面的虚线部分显示的。需要说明的是，操作符”+=”是指将该操作符右边的值“附加”到通过左边的路径得到的特征上。

**步骤3：遍历最后得到的特征结构图，输出最终的RelEx关系集合。**

当步骤2完成后，我们得到的结构特征图包含了很多RelEx句子算法中使用的临时特征，以及残留的句法特征，因此这一步骤主要是遍历步骤2得到的特征结构图并过滤掉这些不必要的特征，得到最终的简洁直观的RelEx关系树（图），最终抽取其中RelEx的依存关系和词语属性集合并输出。这里依然采用上面的例句“The cat chased a snake.”。当所有的句子算法都被遍历过后，再过滤掉临时的特征，可以得到一个精简的用于生成RelEx关系集合的特征结构图（图5.6）。

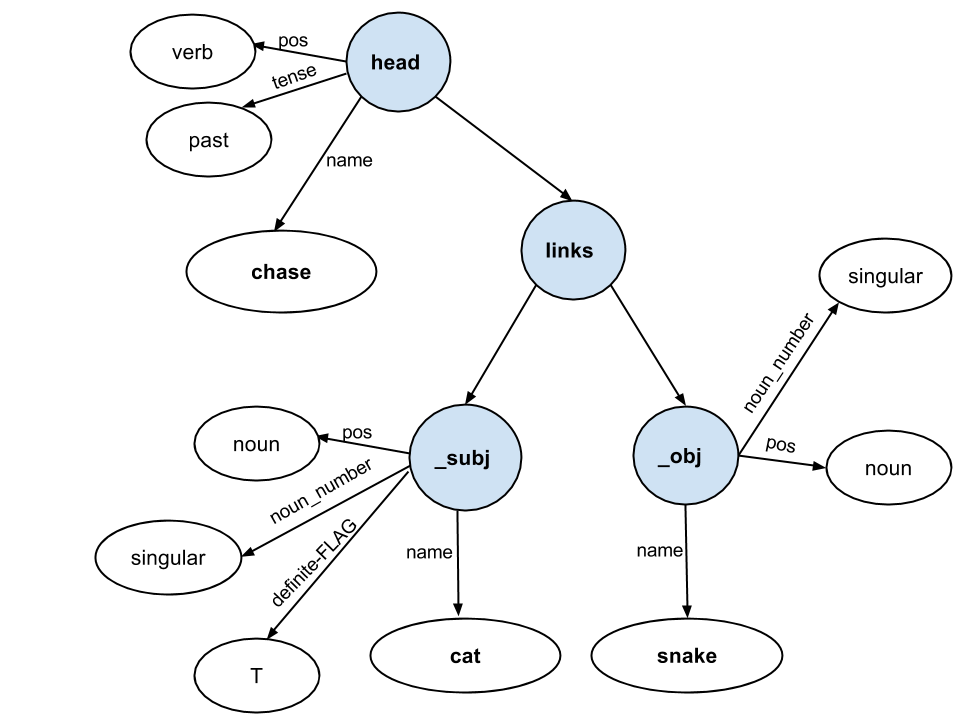


图 5.6 最终特征结构图

其中特征节点head指向表示主句或所有从句中的中心词的特征节点。RelEx中与head对应的还有特征节点background（例句中没有），它是指向表示修辞词的节点，如形容词、副词等。

根据上图，我们可以从以下的特征节点抽取出每个词的相关属性：

* name：指向表示该词的名称的特征节点
* pos：指向表示该词的词性标记的特征节点
* tense：指向表示该词的时态的特征节点（针对动词）
* deﬁnite-FLAG：指向一个值为“T”的特征节点，表明该名词受量词、定冠词等限定词约束，如“the””that”等（针对名词）
* noun\_number：指向表示该词的单复数形式的特征节点（针对名词）

类似，可以从以下的特征节点抽取出相关的依存关系：

* links：指向表示以该节点为第一参数的依存关系的特征节点
* \_subj：指向表示依存关系\_subj（主谓关系）的第二参数的特征节点
* \_obj：指向表示依存关系\_obj（动宾关系）的第二参数的特征节点

根据图5.65.6中的信息，不难抽取最终的RelEx输出结果：

Dependency relations：

\_obj(chase, snake)

\_subj(chase, cat)

Attributes：

tense(chase, past)

subscript-TAG(chase, .v-d)

pos(chase, verb)

pos(., punctuation)

subscript-TAG(snake, .n)

pos(snake, noun)

noun\_number(snake, singular)

definite-FLAG(cat, T)

subscript-TAG(cat, .n)

pos(cat, noun)

noun\_number(cat, singular)

pos(a, det)

pos(the, det)

其中，依存关系（Dependency relations）可以理解成两个词之间语义角色关系，如主谓关系、动宾关系等[[7]](#footnote-8)。属性（Attributes）是值每个词的词态属性，如词性、时态、人称等[[8]](#footnote-9)。

### RelEx 关系的形式化

如前所述，RelEx关系反射了词和词之间的依存关系，如主谓关系、动宾关系、 修饰和被修饰关系、所有格关系等等；除此之外，RelEx 也同样标注了词的各种 词态属性，如时态、词性、单复数等等。为了这些繁琐的关系有一个适合操作以 及后续处理的规范体系，本小节提出了一种基于巴克斯范式 BNF(Backus Normal Form)的对 RelEx 关系进行了如下的形式化表示：

<RelEx-rel>：：== <dependency-rel> | <property-rel>

<dependency-rel>：：== <dependency-type> "(" <word-instance> ", " <word-instance>")"

<property-rel> ：：== <property-type> "(" <word-instance> ", "<property-value> ")"

<word-instance> ：：== <word> <instance> | <word>

其中，

* RelEx-rel可以是RelEx中的表示两个词之间的依存关系，也可以表示单个词的本身属性和属性值之间的关系。
* dependency-rel表示RelEx中的依存关系，其中依次包含依存关系类型（即范式中使用的dependency-type），第一参数（一般是单词，有时候也可以是短语）以及第二参数（一般是单词，有时候也可以是短语）
* property-rel表示RelEx中对词的属性的标注，其中依次包含属性类型（即范式中使用的property-type），词或短语以及属性值。
* word-instance表示来自所分析的句子中的词，如果同一个词在同一个句子中出现2次或以上，通过在后面加上标记（用整数表示）来区分。

## 基于涉身的指代消解

RelEx含有使用典型Hobbs算法实现了简单的指代消解[112]。概括地说，当识别出某语句中的”他”、”她”、”它”等代词，Hobbs算法就会在近期使用过的语句中进行搜索，并根据数字、性别和其他特征来找出匹配此代词的名词。Hobbs算法被用来建立一整列以近因排序的候选名词。

本文在超图知识库Atomspace中实现了一个引入涉身知识的指代消解系统，该系统使用了类似Hobbs算法中的解决思路，但是使用了更灵活的超图知识表示Atomspace，因此能借助超图中的语义链接将知识库中的非语言信息结合起来用于指代消解中。该算法在一个虚拟游戏世界中实现，假设智能体生活在一个周围有很多物体的虚拟环境中，而每个物体都有它们自己的特性，包括该物体的位置、颜色、形状等属性，这样指代消解的任务就能根据上下文所描述的指代物体特性来完成。该实验结果在一段视频中展示[[9]](#footnote-10)。

具体来说，在一游戏世界中，我们可以有许多各种颜色和大小的球。我们运用多个节点将此呈现在Atomspace：表示“球”的概念的单个ConceptNode、与”球”相关联的一个WordNode、以及表示特定球的多个ConceptNode。当然有些ConceptNode也表达其他与球有关的概念，像是”又鼓又大、会发出咯吱声的球”、”可用球棒挥击的球”等不概述于任何自然语言字词的概念。

当智能体与世界互动时，它会通过感知来学到寻获物体的相关信息。关联某设定物体的感知会贮存为其他表示特定物体实例的节点。所有此信息会使用RelEx2Logic被呈现在Atomspace之中（这部分在下一节会举例说明）。

举例来说，当用户说“抓红色的球”(“Grab the red ball”)，智能体就必须理解用户指的是那种球–例如它必须唤起参照消解程序。参照消解使用句中的信息来选择物件、和些许捷思规则。概括而言，参照消解是根据智能体通过感知获取关于世界的知识，将用户句中的名词对应到虚拟世界的实际物体。

在这个例子中，首先智能体会选择相关“球”字眼的ConceptNode，再利用句中的决定因素和其他适当的限制条件，检验所有与这些概念关联的个别实例（就此范例而已，决定因素为形容词“红色”；而由于动词为“抓”，因此它也会寻找可抓取的物体）。若它找到不止一个“可抓取的红球”，就会使用捷思规则来选取一个（在这个情况，智能体会选择离自己最近的物体）。

智能体也须将句中的代词对应到虚拟世界的实际物体。例如，若用户说“我喜欢红色的球。抓起它”(“I like the red ball.Grab it”)，智能体就必须将代词“它”(“it”)对应到特定的红球。此程序分两个阶段执行：首先使用指代消解将代词“它”(“it”)与先前听到的名词“球”(“ball”)联系；接着使用参照消解将名词“球”(“ball”)与实际物体联系在一起。

在此情境中，我们可使用智能体对世界的知识改良Hobbs演算的结果，以帮助选出最佳的候选名词。假设智能体听到这些句子：

"球是红色的。"("The ball is red.")

"棍子是棕色的"("The stick is brown.")

and then it receives a third sentence

"抓它。"("Grab it.")

指代消解器将构建一个列表，此列表包含针对第三个句子中代词“it”的两个 选项：ball 和 stick。如果 stick 是最近使用的名词，那么正如 Hobbs 建议的那样，代 理将抓取 stick，而不是 ball。

与此类似，如果智能体的历史包含

"From here I can see a tree and a ball."

"Grab it."

那么 Hobbs 的算法按顺序返回名词 tree 和 ball 作为候选。但是如果使用我们 的集成的指代消解处理，智能体将推断不能抓取 tree，所以仅选择 ball。

## RelEx2Logic：逻辑关系抽取

本章节介绍我们的自然语言理解系统中最后的纯语言学阶段，即RelEx2Logic。RelEx2Logic是我们本文工作的一个关键组成部分，用于将RelEx生成的句法-语义关系转换成PLN和其它OpenCog认知组件可以使用的基于超图的逻辑表示形式，从而使得我们的自然语言理解系统可以与逻辑推理系统（例如PLN）、控制系统（例如OpenPsi）和OpenCog的其它组件进行有效交互。

### RelEx2Logic

众所周知，自然语言的逻辑极其复杂，而且即使在给定的表示形式下，同一个句子可以用很多种不同的逻辑表达方式，而不影响该句子想要表达的语义。因此在我们的工作中，即通过第二章中介绍的Atomspace来表示自然语言的语义关系和结构，必须面对这个事实：即使对于一个简单地句子，也可以有很多种不同结构的基于Atomspace的逻辑形式来相同或等价的语义。RelEx2Logic系统的目标是以尽可能简单的方式将RelEx输出映射到PLN可以理解和推理的AtomSpace表示，生成可以获取相关表达所包含的大量含义的精确表示。RelEx2Logic不会试图消解RelEx输出中所有的语义歧义，而是专注于将这些语义歧义表达成一种易于发现和处理的方式，这样能使这些歧义义在整个认知体系结构的知识库上或者借助涉身相关的信息，通过简单的常识推理而得到解决。

具体实现上，RelEx2Logic的核心理念是通过应用一系列简单的重写规则以及较复杂的后处理规则（处理复杂句子和问题需要这些后处理规则，例如进行概念实例区分）把RelEx关系映射到语义解释（ AtomSpace表示）。

每个核心重写规则的输入是句法分析图中满足特定约束的子图，输出是一个原子（Atom）超图。在实践中，需要的规则通常采用成对标记(G,A)的形式，其中：

* G是一个图，其节点是单词或变量，其边是RelEx关系类型。
* A是一个超图，其节点是单词、变量或特殊的语言学节点（根据一个小型词汇表绘制），其超图链接是原子（Atom）类型（例如InheritanceLink和EvaluationLink）。
* G和A中的变量列表必须相同。

我们把符合这种说明的规则称为“简单映射规则”。对于处理人类语言真正需要的简单映射规则来讲，满足约束条件G中的每个边映射到A中的单个超图链接。从数学上讲，这个约束条件的后半部分意味着每个重写规则各自都是一个图同态[113]，这进一步意味着一起应用的一系列重写规则也是一个图同态。

这种规则的一个简单例子是这样一个(G,A)，其中

*G* = {*S* ∗(*v*1, *v*2), *O*∗(*v*1, *v*3)}

*A* = (*EvaluationLink v*1 *v*2 *v*3)

这把主语为v2，宾语为v3的动词v1映射到一个v1为谓语，(v2,v3)为参数列表的EvaluationLink。例如，S∗指任何S的链路分析器子类型。当然，大部分规则都比这个例子复杂。

在具体操作中，由于自然语言的复杂性，仍然有一些这种形式的RelEx2Logic规则无法处理的语言现象。所以，为了获取句子正确的逻辑表示而不影响其含义，我们向RelEx2Logic框架添加一个后处理步骤，这个步骤主要包含下列部分：

* 处理Atomspace中的实例。根据输入的自然语言的含义，将通用概念或谓语与特定概念或谓语进行区分。
* 估计模糊谓语或概念的概率。例如，如果句子是“Maybe the cat chased a snake.”，那么我们需要给最后的逻辑表示指定相较于句子“the cat chased a snake.”较低的概率。
* 正确处理量词。众所周知，自然语言中的量词十分复杂。不可能创建一个通用规则而以通用方式获取所有量词的正确抽象表示。而且，我们想要使得我们的规则更加通用，所以我们不想为每个量词使用单独的规则，这需要以简化的方式编写量词规则，而不是极力利用后处理方法允许的更大灵活性。

下面进一步将解释后处理的一些处理细节。需要说明的是，在我们的知识表示体系Atomspace中，系统会为每个节点随机产生一个唯一的序列号，以便用于检索和区分具有相同节点内容的不同节点。由于篇幅关系，我们这里给出的实验结果简化了每个节点的序列号，如系统产生的节点为“chased@64f680a3-d076-4ae1-814c-0152c3f3c8f8”，在本文中会被简化为“chased@64f”。

将我们当前的简单映射规则应用于上述的例句“The cat chased a snake.”，会生成以下基本输出：

EvaluationLink

PredicateNode chased@64f

ListLink

ConceptNode cat@3a1

ConceptNode snake@b8a

这些“基本输出”仅仅是这个句子解释后Atomspace中所创建的完整原子（Atom）集合的一小部分。一个更完整的列表是

((ReferenceLink

(InterpretationNode "sentence@f20\_parse\_0\_interpretation\_$X") (SetLink

(ImplicationLink

(PredicateNode "chased@64f") (PredicateNode "chase" (ptv 1 0 1))

)

(InheritanceLink (ConceptNode "cat@3a1")

(ConceptNode "cat" (ptv 1 0 1))

)

(InheritanceLink (ConceptNode "snake@b8a")

(ConceptNode "snake" (ptv 1 0 1))

)

(EvaluationLink

(PredicateNode "chased@64f") (ListLink

(ConceptNode "cat@3a1") (ConceptNode "snake@b8a")

)

)

(InheritanceLink

(PredicateNode "chased@64f") (ConceptNode "past")

)

(InheritanceLink

(InterpretationNode "sentence@f20\_parse\_0\_interpretation\_$X") (ConceptNode "DeclarativeSpeechAct")

)

(EvaluationLink (PredicateNode "definite") (ListLink

(ConceptNode "cat@3a1")

)

)

)

)

本节最后会给出更多映射例子以及所涉及原子（Atom）类型的解释。

如果链路分析器给出多个分析选项，RelEx2Logic 规则将为每个分析选项提供 一个逻辑解释。目前已经实现基于语料库进行统计启发来排列多个分析的可信性， 当然这并不完美。一些情况中，会向 OpenCog 提供多个逻辑输出选项，并且必须 基于更高层次的上下文推断在这些选项之间进行选择，这是当前研究的一个难题和 主题。

### RelEx2Logic 所使用的规则

为了将RelEx关系转换成抽象的OpenCog AtomSpace逻辑表示，我们引入一系列RelEx2Logic规则。RelEx2Logic规则（以前称为Link2Atom和Syn2Sem规则[]）是一系列人工编码的规则，用于把RelEx的输出映射到OpenCog原子（Atom）表示的形式，这种表示更适合于由PLN和其它工具进行自动化推断。本节总结性地阐述规则集合并给出一些规则例子。可以在opencog/nlp/scm/relex-to-logic.scm在线找到所有规则。

RelEx2Logic规则有多个目标：

* 作为运行的OpenCog自然语言处理系统的一部分，其潜在用途包括文本分析和自然语言对话。
* 作为概念研究工具，帮助我们理解句法-语义输出（例如RelEx生成的句法-语义输出）映射到更抽象的逻辑关系（例如PLN采用的逻辑关系）的问题。
* 作为工具来帮助生成成对（链路分析和抽象的PLN形式的原子）的语料库，然后可以自动分析此语料库来产生语言理解和生成规则（更多信息请参见Syn2Sem）。

**示例规则（Tense规则）**

为了方便阐述剩下的内容，我们从一个非常简单的RelEx2Logic规则例子开始。当然，大部分规则比这个例子复杂。此处，为了方便人理解，采用RelEx SFF格式（RelEx许多输出格式中的一种）来说明规则输入。

为了用便于阅读的方式说明RelEx2Logic规则，我们在此使用类似以下的格式：

规则输入：

tense(W, Tense) pos(W, verb)

规则输出：

(tense-rule W (get\_instance\_name W word\_index sentence\_index) Tense)

在OpenCog中运行此规则，规则的输出将启动一个帮助函数（helper function，用Scheme语言编写）。下面会列出这个帮助函数。然后帮助函数将最终生成最后的OpenCog Atomspace表示。

如同上一小节中所述，在我们软件实现中，我们使用诸如(ImplicationLink PG PA)这样的格式来表示RelEx2Logic规则，从而使得RelEx2Logic与OpenCog系统的其余组件保持一致。例如，这种格式意味着规则可以由OpenCog模式匹配器来执行（OpenCog模式匹配器用来在OpenCog Atomspace中搜索原子的特定模式、排列或“模板”）。所以上述简单的RelEx2Logic规则在当前系统中将被如下表示：

ImplicationLink

EvaluationLink

PredicateNode "tense"

ListLink

WordNode W

ConceptNode Tense

ExecutionOutputLink

GroundedSchemNode "r2l/tense-rule.scm"

ListLink

WordNode W

ConceptNode Tense

NumberNode word\_index

NumberNode sentence\_index

对应于上述 RelEx2Logic 规则的提供后处理所需要的相应的辅助函数为：

辅助函数为：

(define (tense-rule verb instance tense)

(define new\_predicate (PredicateNode instance))

(define verb\_node (PredicateNode verb))

(define tense\_node (ConceptNode tense))

(list

(InheritanceLink new\_predicate verb\_node)

(InheritanceLink new\_predicate tense\_node)

)

)

例如：

tense(chase, Past) pos(chase, verb)

==>

(tense-rule "chase" "chased@64f" "Past")

==>

(InheritanceLink "chased@64f" chase)

(InheritanceLink "chased@64f" Past)

注意：这个例子使用了额外的帮助函数get\_instance\_name。下面解释这个帮助函数。本质上，这个函数只是为概念的实例选择一个在给出的AtomSpace中唯一的名字（例如，它可能为chair的实例选择名字chair77）。如果单词W在句子中是第K次出现，那么单词序号为K。句子序号是该句子区别于其它句子的一个标识符。

### RelEx2Logic的后处理过程

RelEx2Logic最初的设计意图是如同上述例子那样通过直接的逻辑含义进行所有需要的处理。但是随着工作的进展，我们认识到这样做在许多情况中会非常麻烦，同时我们发现引入一些更精巧的后处理方法是最直接的解决办法。此处我们阐述这些后处理规则的基本类型。

现阶段，我们已经实现的后处理规则包括下列类别：

* 量词：为带有量词的单词引入变量并通过使用相应的链路限定变量的范围，同时给链路指定一些真值。
* 模糊情态动词：为模糊情态动词发生过修改的谓语指定一些置信下限。例如“Maybe dogs likeﬁsh.”，这表示“dogs likeﬁsh”的置信度不是很高，所以我们把“he is a genius.”的置信度从默认的0.99调整为0.5。
* 冗余实例清理：为可以从句法分析知道的通用概念清理实例号。例如“Dogs like meat.”，没有为dogs和meat指定任何明确的冠词、修饰语或代词，并且也没有意指任何其它概念，所以我们认为dogs和meat指通用概念dog和meat，而不是某些特定概念（特定概念需要指定实例号）。

下面更详细地阐述我们怎样为上面列出的每种类别实现后处理。

**量词：**在前面的阶段，我们使用量词记号来标记需要进行后处理的量词。例如“All dogs like meat.”， 在应用量词RelEx2Logic规则期间创建 “allmarker”并生成下列结果。

EvaluationLink

PredicateNode "allmarker"

ListLink

ConceptNode "dog@13456"

本例中，在后处理阶段，我们为量词“all”引入“ForAllLink”（它代表OpenCog Atomspace中的通用量化）和变量节点“$X”来限定范围包含所有含有dog@13456的链路。然后，每个dog@13456都将被替换为VariableNode“$X”。“allmarker”将在后处理后被删除。所以，在后处理结束后，将创建：

ForAllLink

VariableNode "$X" ImplicationLink

InheritanceLink "$X" noun\_instance

AndLink

\*\* links involving "dog@13456" \*\*

**模糊情态动词：**模糊情态动词后处理类似于量词后处理，我们在应用相关 的 RelEx2Logic 规则期间使用相应的记号来标记模糊情态动词。例如，对于句子 “Maybe dogs like ﬁsh.”，在应用 Maybe RelEx2Logic 规则后生成下列表示：

EvaluationLink

PredicateNode "maybemarker" ListLink

PredicateNode "like@9768"

然后，用于后处理的maybemarker帮助函数将查找所有包含“like@9768”的根链路，并将置信度从默认的0.99调整为0.5。

**冗余实例清理：**因为不想在句子转换成抽象语义表示期间丢失任何来自句法层面的信息，我们在后处理阶段设立“实例清理”步骤。在当前方法中，我们为句子中每个非组合单词创建一个单独的实例节点。但是，在很多情况中，这会生成许多不必要的实例节点。例如，在句子“Dogs like yummy meat.”中，无需创建诸如yummy@1765和dog@2593的实例节点，因为这个句子没有指定特定的“dog”和特殊的“yummy”。所以只创建类似以下的链路就已足够：

InheritanceLink

ConceptNode "meat@3976" ConceptNode "yummy"

EvaluationLink

PredicateNode "like" ListLink

ConceptNode "dog"

ConceptNode "meat@3976"

但是，一个诸如“meat@3976”的实例节点更加有用。因为在这个句子中，它仅陈述“Dogs like’YUMMY’meat”，而并非一般的“meat”。

要想知道哪些单词真正需要实例节点，必须在一定程度上理解句子的结构。当前，我们使用一些常识性的规则开始这个过程。确切地讲，如果一个单词没有被修改（名词被形容词修改或者动词被副词修改），或者没有被定冠词限定，或者使用代词，那么此单词的实例节点为“冗余”。清理实例的后处理进行以下工作：

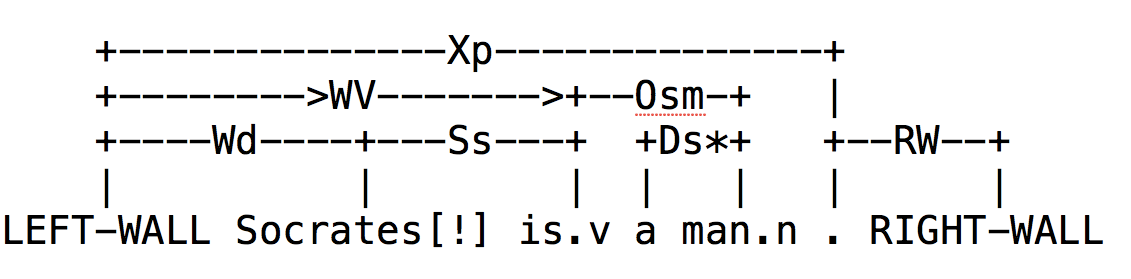
* 在Atomspace中（或者仅在RelEx2Logic规则的本地输出中）检查是否存在任何此名字的原子为“冗余”。
* 如果找到冗余原子，那么将其删除并将它的链路移至其父概念。

## 实验结果及分析

本节中给出更多例子来解释我们的语言理解的理解管道，包含前面阐述的所有部分：Link Parser、RelEx和RelEx2Logic。在后面的章节中，这些特定例子也会在下一章中用来阐述基于自然语言的推理。

**例1：Socrates is a man**

链语法分析器输出：



RelEx 输出：

Dependcy Relations：

\_obj(be, man)

\_subj(be, Socrates)

Attributes：

tense(be, present)

subscript-TAG(be, .v)

pos(be, verb)

pos(., punctuation)

subscript-TAG(man, .n)

pos(man, noun)

noun\_number(man, singular)

definite-FLAG(Socrates, T)

pos(Socrates, noun)

noun\_number(Socrates, singular)

pos(a, det)

**RelEx2Logic 输出：**

((ReferenceLink

(InterpretationNode "sentence@f04\_parse\_0\_interpretation\_$X") (SetLink

(InheritanceLink

(ConceptNode "Socrates@cfb") (ConceptNode "Socrates" (ptv 1 0 1))

)

(InheritanceLink (ConceptNode "man@fe4")

(ConceptNode "man" (ptv 1 0 1))

)

(InheritanceLink

(ConceptNode "Socrates@cfb") (ConceptNode "man@fe4")

)

(ImplicationLink (PredicateNode "is@b06")

(PredicateNode "be" (ptv 1 0 1))

)

(InheritanceLink (PredicateNode "is@b06") (ConceptNode "present")

)

(InheritanceLink

(InterpretationNode "sentence@f04\_parse\_0\_interpretation\_$X") (ConceptNode "DeclarativeSpeechAct")

)

(EvaluationLink (PredicateNode "definite") (ListLink

(ConceptNode "Socrates@cfb")

)

)

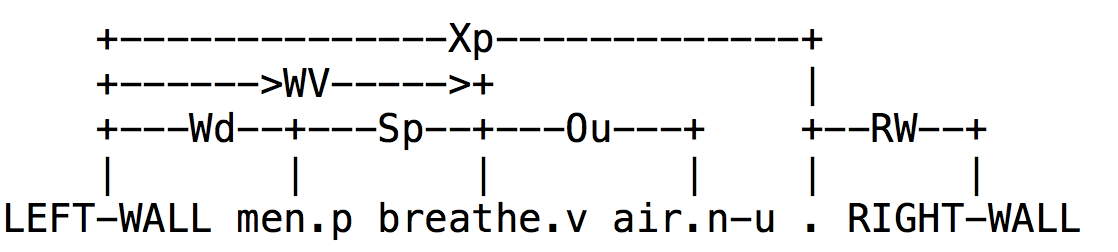
)

)

)

**例 2： Men breathe air**

链语法分析器输出：



RelEx 输出：

Dependcy Relations：

\_obj(breathe, air)

\_subj(breathe, man)

Attributes：

tense(breathe, present)

subscript-TAG(breathe, .v)

pos(breathe, verb)

pos(., punctuation)

subscript-TAG(air, .n-u)

pos(air, noun)

noun\_number(air, uncountable)

subscript-TAG(man, .p)

pos(man, noun)

noun\_number(man, plural)

RelEx2Logic 输出：

((ReferenceLink

(InterpretationNode "sentence@05b\_parse\_0\_interpretation\_$X") (SetLink

(ImplicationLink

(PredicateNode "breathe@ba9")

(PredicateNode "breathe" (ptv 1 0 1))

)

(InheritanceLink (ConceptNode "men@5f6")

(ConceptNode "man" (ptv 1 0 2))

)

(InheritanceLink (ConceptNode "air@98a")

(ConceptNode "air" (ptv 1 0 1))

)

(EvaluationLink

(PredicateNode "breathe@ba9") (ListLink

(ConceptNode "men@5f6")

(ConceptNode "air@98a")

)

)

(InheritanceLink

(PredicateNode "breathe@ba9") (ConceptNode "present")

)

(InheritanceLink

(InterpretationNode "sentence@05b\_parse\_0\_interpretation\_$X") (ConceptNode "DeclarativeSpeechAct")

)

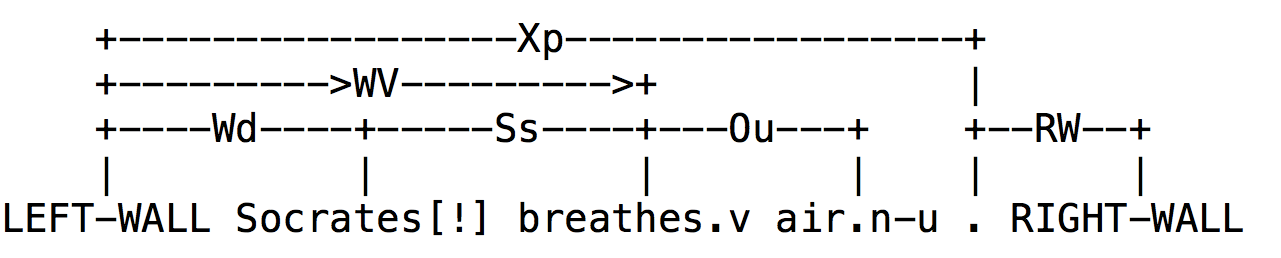
)

)

)

**例 3： Socrates breathes air**

链语法分析器输出：



RelEx 输出：

Dependcy Relations：

\_obj(breathe, air)

\_subj(breathe, Socrates)

Attributes：

tense(breathe, present)

subscript-TAG(breathe, .v)

pos(breathe, verb)

pos(., punctuation)

subscript-TAG(air, .n-u)

pos(air, noun)

noun\_number(air, uncountable)

definite-FLAG(Socrates, T)

pos(Socrates, noun)

noun\_number(Socrates, singular)

RelEx2Logic 输出：

((ReferenceLink

(InterpretationNode "sentence@a41\_parse\_0\_interpretation\_$X") (SetLink

(ImplicationLink

(PredicateNode "breathes@0a0") (PredicateNode "breathe" (ptv 1 0 2))

)

(InheritanceLink

(ConceptNode "Socrates@a34") (ConceptNode "Socrates" (ptv 1 0 2))

)

(InheritanceLink (ConceptNode "air@016")

(ConceptNode "air" (ptv 1 0 2))

)

(EvaluationLink

(PredicateNode "breathes@0a0") (ListLink

(ConceptNode "Socrates@a34")

(ConceptNode "air@016")

)

)

(InheritanceLink

(PredicateNode "breathes@0a0") (ConceptNode "present")

)

(InheritanceLink

(InterpretationNode "sentence@a41\_parse\_0\_interpretation\_$X") (ConceptNode "DeclarativeSpeechAct")

)

(EvaluationLink (PredicateNode "definite") (ListLink

(ConceptNode "Socrates@a34")

)

)

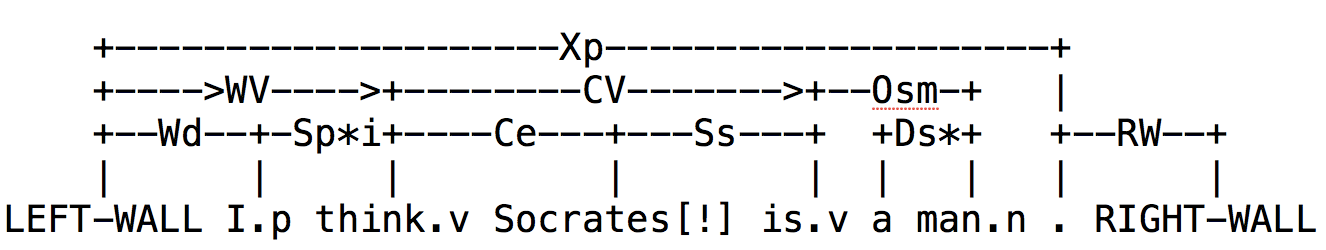
)

)

)

**例 4： I think Socrates is a man**

链语法分析器输出：



RelEx 输出：

Dependcy Relations：

\_obj(be, man)

\_subj(be, Socrates)

\_rep(think, be)

\_subj(think, I)

Attributes：

definite-FLAG(Socrates, T)

pos(Socrates, noun)

noun\_number(Socrates, singular)

pos(., punctuation)

subscript-TAG(man, .n)

pos(man, noun)

noun\_number(man, singular)

pos(a, det)

tense(be, present)

HYP(be, T)

subscript-TAG(be, .v)

pos(be, verb)

tense(think, present)

subscript-TAG(think, .v)

pos(think, verb)

pronoun-FLAG(I, T)

gender(I, person)

definite-FLAG(I, T)

subscript-TAG(I, .p)

pos(I, noun)

noun\_number(I, singular)

RelEx2Logic 输出：

((ReferenceLink

(InterpretationNode "sentence@ec1\_parse\_0\_interpretation\_$X") (SetLink

(ImplicationLink (PredicateNode "think@8be")

(PredicateNode "think" (ptv 1 0 1))

)

(InheritanceLink (ConceptNode "I@8c8")

(ConceptNode "I" (ptv 1 0 1))

)

(EvaluationLink

(PredicateNode "think@8be") (ListLink

(ConceptNode "I@8c8")

)

)

(ImplicationLink (PredicateNode "is@14df")

(PredicateNode "be" (ptv 1 0 2))

)

(EvaluationLink

(PredicateNode "think@8be") (ListLink

(ConceptNode "is@14d")

)

)

(InheritanceLink (PredicateNode "think@8be") (ConceptNode "present")

)

(InheritanceLink

(ConceptNode "Socrates@c1f") (ConceptNode "Socrates" (ptv 1 0 3))

)

(EvaluationLink (PredicateNode "definite") (ListLink

(ConceptNode "Socrates@c1f")

)

)

(InheritanceLink

(InterpretationNode "sentence@ec1\_parse\_0\_interpretation\_$X")

(ConceptNode "DeclarativeSpeechAct")

)

(InheritanceLink (ConceptNode "man@aed")

(ConceptNode "man" (ptv 1 0 3))

)

(InheritanceLink

(ConceptNode "Socrates@c1f") (ConceptNode "man@aed")

)

(InheritanceLink (PredicateNode "is@14d") (ConceptNode "present")

)

(EvaluationLink (PredicateNode "definite") (ListLink

(ConceptNode "I@8c8")

)

)

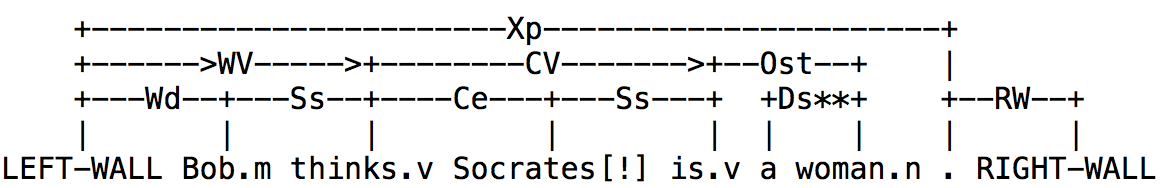
)

)

)

**例 5： Bob thinks Socrates is a woman**

链语法分析器输出：



RelEx 输出：

dependcy Relations：

\_obj(be, woman)

\_subj(be, Socrates)

\_rep(think, be)

\_subj(think, Bob)

Attributes：

definite-FLAG(Socrates, T)

pos(Socrates, noun)

noun\_number(Socrates, singular)

pos(., punctuation)

subscript-TAG(woman, .n)

pos(woman, noun)

noun\_number(woman, singular)

pos(a, det)

tense(be, present)

HYP(be, T)

subscript-TAG(be, .v)

pos(be, verb)

tense(think, present)

subscript-TAG(think, .v)

pos(think, verb)

gender(Bob, masculine)

definite-FLAG(Bob, T)

person-FLAG(Bob, T)

subscript-TAG(Bob, .m)

pos(Bob, noun)

noun\_number(Bob, singular)

RelEx2Logic 输出：

((ReferenceLink

(InterpretationNode "sentence@8c6\_parse\_0\_interpretation\_$X") (SetLink

(ImplicationLink

(PredicateNode "thinks@87f") (PredicateNode "think" (ptv 1 0 2))

)

(InheritanceLink (ConceptNode "Bob@f16")

(ConceptNode "Bob" (ptv 1 0 1))

)

(EvaluationLink

(PredicateNode "thinks@87f") (ListLink

(ConceptNode "Bob@f16")

)

)

(ImplicationLink (PredicateNode "is@712")

(PredicateNode "be" (ptv 1 0 3))

)

(EvaluationLink

(PredicateNode "thinks@87f") (ListLink

(ConceptNode "is@712")

)

)

(InheritanceLink

(PredicateNode "thinks@87f") (ConceptNode "present")

)

(InheritanceLink

(ConceptNode "Socrates@ead") (ConceptNode "Socrates" (ptv 1 0 4))

)

(EvaluationLink (PredicateNode "definite") (ListLink

(ConceptNode "Socrates@ead")

)

)

(InheritanceLink

(InterpretationNode "sentence@8c6\_parse\_0\_interpretation\_$X") (ConceptNode "DeclarativeSpeechAct")

)

(InheritanceLink (ConceptNode "woman@9d7")

(ConceptNode "woman" (ptv 1 0 1))

)

(InheritanceLink

(ConceptNode "Socrates@ead") (ConceptNode "woman@9d7")

)

(InheritanceLink (PredicateNode "is@712") (ConceptNode "present")

)

(InheritanceLink (SpecificEntityNode "Bob@f16") (ConceptNode "male")

)

(InheritanceLink (SpecificEntityNode "Bob@f16") (ConceptNode "Bob" (ptv 1 0 1))

)

(EvaluationLink (PredicateNode "definite") (ListLink

(ConceptNode "Bob@f16")

)

)

)

)

)

## 比较级的处理：实例分析

比较级提供了从表层形式映射到我们设计和实现的逻辑表示的有趣例子。关于正确处理英语和其它语言中的比较级，理论上的语言学从来没有达成一致。一些理论家假定一种省略理论，建议从句子的表层结构得出比较级语法，忽略深层结构中存在的某些单词[115，116]。另外一些理论家假定一种移动理论[117，118]，这种理论更多地由传统的生成语法启示，假设比较级语法包含一个重新排列深层结构的表层结构。

链路语法框架从根本上回避了这种问题：不管省略理论还是移动理论，都由链路语法词典中的某些对称性表示，但是这些对称性不需要被链路分析器本身显式地识别或使用（虽然这些对称性可以指导人类或AI系统创建链路语法词典）。从目前的经验上讲，链路分析器对比较级的处理相当好，但是相关的词典条目有点混乱并且实际上并不完全对称。这说明两种可能：

1. 英语比较级的语法“复杂”而混乱，不适用任何可用的理论。并且/或者
2. 链路语法词典可以在比较级方面进行极大的改进

我们猜测事实是两者兼而有之。但是请注意，将链路语法作为用于理解复杂句子（包含比较级）的实际管道的一部分而部署时，我们不需要决定这个问题。

我们通过一个例子说明我们的框架如何处理比较级。RelEx2Logic的一个用于比较级的规则，其简短形式如下：

than(w1, w2)

\_comparative(ad, w)

==>

TruthValueGreaterThanLink

InheritanceLink w1 ad

InheritanceLink w2 ad

图[5.7](#_bookmark88)给出了完整形式。

使用这个规则的一个简单例子是：

Pumpkin is cuter than the white dog.

==>

\_predadj(cute, Pumpkin)

than(Pumpkin, dog)

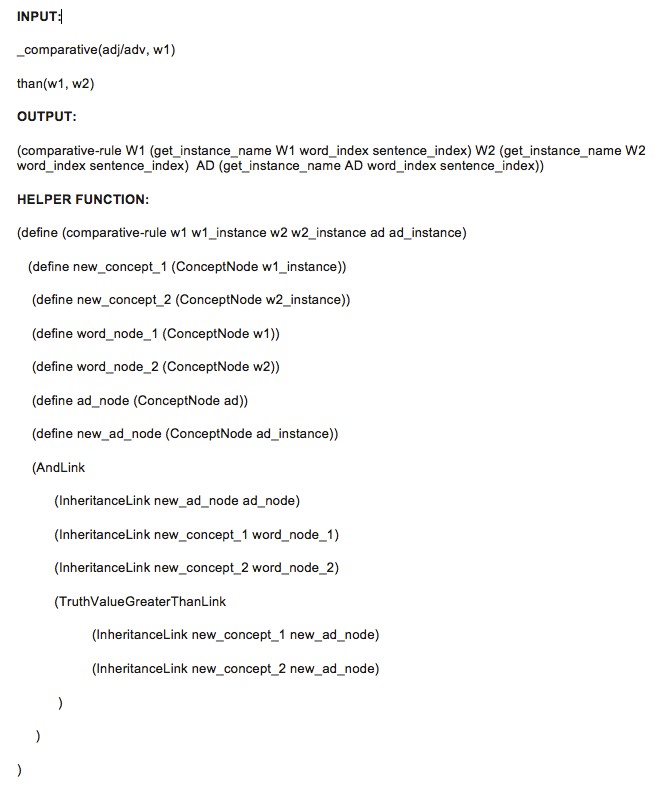


图 5.7 Full RelEx2Logic rule for parsing one type of comparative sentence.

\_comparative(cute, Pumpkin)

\_amod(dog, white)

==>

AndLink

InheritanceLink dog\_11 white

InheritanceLink dog\_11 dog

TruthValueGreaterThanLink

InheritanceLink Pumpkin cute

InheritanceLink dog\_11 cute

另一方面，为了处理诸如“Amen is more intelligent than insane”这样的句子，我们使用一个不同的规则，这个规则的简短形式如下：

\_predadj(adj1, W) than(adj1, adj2)

\_comparative(adj1, W)

==>

TruthValueGreaterThanLink

InheritanceLink W adj1

InheritanceLink W adj2

结果输出如下：

\_predadj(intelligent, Amen)

than(intelligent, insane)

\_comparative(intelligent, Amen)

==>

TruthValueGreaterThanLink

InheritanceLink Amen intelligent

InheritanceLink Amen insane

## 在简单句子集的性能评价

由于本文中的自然语言理解系统最终输出的基于超图逻辑形式比较独特，无法与其他类似系统进行比较。这里我们从路透社的语料库中随机抽取了100个词数小于8的句子作为测试集（选择相对较短句子来作为测试集，我们的出发点在于着重测试该自然语言理解系统对词语或短语之间的语义关系，而非专注于处理复杂的语言现象），在上面运行了我们的自然语言处理系统，并对输出结果做了人工分析，评价结果可参见下表：

表5.1 改进后的链语法分析器在该测试集上的分析结果。“几乎正确”是指分析结果中忽略了链语法词典中不存在的词，但对剩下部分解析正确。

|  |  |
| --- | --- |
|  | Link Parser |
| 完全正确  几乎正确  错误 | 96  2  2 |

表5.2 在链语法分析器解析完全正确的96个句子上，RelEx的性能分析。其中“几乎正确”是指输出中忽略或错误输出了1个以内的依存关系。

|  |  |
| --- | --- |
|  | RelEx |
| 完全正确  几乎正确  错误 | 80  15  1 |

表5.3 在链语法分析器和RelEx都解析完全正确的80个句子上，RelEx2Logic的性能分析。其中“几乎正确”是指输出中忽略或错误输出了1个以内的语义关系。

|  |  |
| --- | --- |
|  | RelEx2Logic |
| 完全正确  几乎正确  错误 | 68  12  0 |

在该测试集中，链语法解析器唯一解析完全失败的句子是：

But can one, really?

在该测试集中，很多改进后的链语法解析器没有完全解析正确的句子，主要是因为对习语的分析错误，如：

It has held them at bay.

（这里的“at bay’并不是字面意义的“bay”’）。

在该测试集中，RelEx 分析完全失败的也只有 1 句：

Part of it is, of course

其输出如下：

Error： No target! rel=\_pobj and src=of

\_advmod(be, of\_course)

\_subj(be, it)

\_quantity(it, part)

\_det(it, part)

\_advmod(part, of)

但也有一些情况，RelEx忽略了很重要的依存关系，如：

The expression has become quite a cliche.

输出中只有下列依存关系：

\_obj(become, cliche)

\_subj(become, expression)

同时还有下列属性：

pos(quite\_a, det)

idiom-FLAG(quite\_a, T)

该例中，RelEx忽略了“quite a”和”cliche’之间的依存关系；类似，在该测试集上，RelEx2Logic也几乎都给出合理的语义关系，但有时的输出也不完整。如：

But there is also compassion.

RelEx2Logic没有产生与“but”相关的语义关系，这是因为有关“but”的RelEx2Logic 规则还没写入规则库it does not produce a relation for “but” because at this time no “but” rule has been coded. Similarly in

总的来说，在该简单的测试集上的结果是鼓舞人心的，它能说明，本章阐述的自然语言系统在简单句子上的表现性能是不错的，其中大部分错误也都能通过添加相关的规则来解决。当然，基于规则的系统的本质也在于能通过不断添加和修改规则来提高系统的性能，但总归是很耗时的，而且也很不灵活，因此，我们希望能使机器自动从无标注或者半标注的语料库中学习相关的语法规则来替换我们所使用的人工编写的规则库。目前我们已着手研究无监督的语言学习的算法[[10]](#footnote-11)，但这并不是本文的重点，因此不在这里介绍。

## 本章小结

本章主要介绍我们的自然语言理解系统中的各个模块的工作原理和实现方法，并深入讨论了该系统如何将英文句子转换成超图表示的逻辑形式，以达到能使自然语言参与到逻辑推理的过程中。该自然语言理解系统已经被应用在游戏角色控制和人形机器人等领域。

该自然语言理解系统使用了一些现有的开源语法分析器，本章也讨论对这些开源工具做出的各种改进，还讨论了我们实现的全新的将依存关系转换成语义逻辑表示的系统RelEx2Logic，总的来说，本章的贡献有：

* 改进了链语法分析器的词典，使其能够处理更复杂的语言现象，以及能识别句子的中心词从而改进了链语法对复句的分析能力。
* 改进了RelEx使其能处理目前语法分析器难以处理的比较级和量词等复杂的语言现象，并将RelEx中的指代消解算法用到Atomspace中实现了一个引入涉身知识的指代消解系统。
* 设计并实现了RelEx2Logic系统，使其能将RelEx输出的依存关系转换成以超图表示的逻辑形式。
* 设计并实现了RelEx2Logic中所使用的转换规则及相应的规则引擎。

总的来说，本章搭建的自然语言理解系统已经能被用在一些用户定义的实际应用中，但仍然有很多遗留问题有待解决。在接下来的工作中，我们会在我们的自然语言理解系统中引入概率逻辑推理引擎PLN，这样不仅可以通过一些基本常识推理来改进语法分析中的结果，还可用于指导对语法分析器产生的多种结果进行排序，以及用于指导各种歧义消除和指代消解等方面。最后，本章还在一个小的测试集上对我们的自然语言理解系统进行了测试并做了相应的评价分析。

# 

第六章 自然语言生成：从逻辑到语言

与第[5](#_bookmark66)章阐述的自然语言理解系统的工作（将英语句子转换成超图表示的逻辑 形式）正好相反，本章将要的阐述的自然语言生成系统的工作是将基于超图表示的 逻辑结构转换成合法的英语句子。本章的研究工作主要针对第[1](#_bookmark4)章中的提到的假设 2 展开。根据该假设，我们认为，借助一个超图转换系统和一个超图匹配系统，在 由自然语言理解系统自动生成的二元组（语言表达式，逻辑表达式）组成的知识库 中根据逻辑表达式找到匹配的语言表达式并生成自然语言，是可行的。

基于这样的假设，我们在自然语言生成系统中设计并实现了一个超图转换系 统，用于自然语言生成中的微观规划（Microplanning）；和一个超图匹配系统，用 于自然语言生成中的表层生成（Surface Realization）。我下面章节将分别对这两个 子系统的工作原理和实现方法等做进一步地讨论。



## 微观规划器

本章节主要介绍我们的自然语言生成系统中的用于微观规划的子系统，我们称其为微观规划器（以下称Microplanner）。

从我们的要实现长期目标即认知对话系统的设计角度来看，对话管理模块输出的基于超图表示的逻辑表示，这些逻辑表示包含了认知对话系统想要表达的内容，即“说话内容”，说话内容的逻辑表示只是对话管理模块在没有相关的语法约束，根据当时的语境从知识库Atomspace中收集到的有关信息，因此，这些表示”说话内容“的超图表示可能无法直接通过在（语言表达式，逻辑表达式）的二元组的知识库中找到相应的语言表达式来生成句子。因此，我们引入了Microplanner。Microplanner的任务就是接收从对话系统传来的表示”说话内容“的超图表示，通过超图转换，生成能够符合表达要求的超图。

从实现角度来看，Microplanner的工作在OpenCog的统一知识库Atomspace中进行，主要任务是将一组Atoms（用S表示）和一些用于表示话语类型（如祈使句、疑问句等）的参数（该参数可以从第4.3节介绍的言语行为规划器中获得4.3），在不改变这些S中所包含的”说话内容“的前提下，转换成符合输入参数所需的话语类型的，且符合英语语法的一组新的Atoms（用S’表示）。需要说明的是，为了能规划出一个完整的句子，Microplanner也有可能使用S之外的Atom。在实现过程中，我们利用Atomspace中SetLink（参见第2.1.2节中的Atomspace的基本边和节点）来连接S中的所有Atoms。

目前，我们的Microplanner还不够成熟，但已经实现了基本的功能。其中包括以下话语类型的表达：

* 陈述句（Declarative）：对于陈述句话语类型，Microplanner的目标就是能表明S中的所含的内容且符合语法的超图表示S‘
* 疑问句（Interrogative）：对于疑问句话语类型，Microplanner的目标可以是下面两项：
* 对一般疑问句，征求某一个超图的边L的真值(L包含S中的内容)
* 对特殊疑问句，征求可以填充某个变量节点VariableNode$V的内容（S构成$V的约束）
* 祈使句（Imperative）：对于祈使句话语类型，其目标是产生一组Atoms，使其包含S中希望听话人能服从的命令内容。
* 感叹句（Interjective）：对于感叹句话语类型，其目标就是能表达S中的相关语用内容

Microplanner的相应算法流程可表示如下：

MakeUtterance(Atom-Set S, utterance type t)

\\ the utterance type t = declarative, interrogative, imperative, etc.

\\ 如果话语类型输入为空，那么系统会考虑所有的类型。

Initialize S\_leftover=S

Repeat until S\_leftover = empty ：

Make a sentence by calling MakeSentence(S\_leftover, S, t)

Let S\_used = the Atoms from S used inside the

above invocation of MakeSentence (i.e. the Atoms expressed in the sentence that MakeSentence produces)

Let S\_leftover = S\_leftover - S\_used

其中，针对陈述句类型的 MakeSentence 流程可表示如下（其他话语类型类 似）：

MakeSentence(Atom-Set S\_available, S\_just\_said, utterance type t)

S\_available = the Atoms available for utterance, within S

S\_just\_said = the Atoms just said within the same utterance, useful for inserting anaphora

Initialize S\_leftover=S\_available

Pick a top-level sentence form (e.g. for declarative it could be SV or SVO).

Pick a set S\_start consisting of Atoms in S\_leftover that match the top-level sentence form chosen

(i.e. that match the output of the RelEx2Logic rule corresponding to the top-level sentence form)

Let S\_working = S\_start

If S\_working contains Atoms from S\_just\_said, or from previous versions of S\_working within this invocation of MakeSentence, then consider inserting anaphora to refer to them

(\*\*) Use SuReal (and a reverse of the chosen RelEx2Logic rule) to produce a sentence corresponding to S\_working

If the sentence produced is too long or complex, DONE

(i.e., decide that the sentence is done and additional Atoms must go into a new sentence, referring back to the current one)

Now, pick a RelEx2Logic rule whose output matches some set S\_new of Atoms that overlaps with S\_working

Let S\_working = S\_working + S\_new

Goto step \*\*

DONE

目前，Microplanner 的工作主要通过两个模块来完成：组块和指代的引入。下 面将具体介绍这两个模块的实现方法。

### 组块

从上面的Microplanner算法的伪代码可得知，Microplanner首先选择输入S中的子集来进行组块，然后不断迭代，直到输入S中所有内容都被规划。组块的算法可描述如下：

1. 对输入的超图（S的子集）中的每条边根据以下几个权重因子来排序：

* Form- weight (0…1)：该边是否能满足基本句式（Microplanner根据话语类型定义了相应的基本句式）
* time- weight (n…0)：集合中的时间顺序（很多表达输出需要考虑时间顺序，
* 比如“我要去厦门参加论文答辩”中“去厦门”和“参加论文答辩”必须按照一定顺序输入）
* link-weight(n)：当前输入的超图中与已经规划过的内容中相同节点的个数。

其中默认的权值计算公式如下：*form− weight∗(time− weight+ link− weight)* 。

1. 选择权值最高的边，调用SuReal（在下一节中会讨论），判断该边是否是“可表达的”（即能在知识库中找到相应的语言表达式），或者是否是“很复杂的”（即该组块能在知识库中找到超过n种话语类型的语言表达式，n默认为3）。其中“很复杂的”因素只是针对输入时没有指定话语类型的情况。
2. 如果该组块是“可表达的”，但是还能更长，那么进行以下操作：

* 对剩下的超图中的所有边重新排序，但这次排序的目的在于选择不同于已使用的句式的边A。这样考虑是因为，如果已使用的句式不能完全表达A中的内容，那么可以试试其他的句式。
* form-weight (0…1)：同上
* time-weight (n…0)：同上
* link-weight (n)：当前的超图中与即将说的内容中的相同节点的个数

其中默认的权值计算公式如下：*(time−weights+link−weights)∗(2−f orm−weights)*

* 选择权值最高的边，然后继续调用SuReal试着再说一遍。

1. 如果该组块不是“可表达的”，那么：
   * 检查当前组块，看是否有节点只出现过1次
   * 在该节点对应的边加入组块中
   * 继续这样尝试n次（n默认为3）后，如果该组块还不是“可表达的”那么放弃。

### 指代的引入

为了让句子的表达更具语言色彩以及更符合人类的交流习惯，还需要对已经描述的对象进行指代处理，也就是利用代词、固定名、完整或缩略名词短语来替换那些已经被描述过的对象，因此Microplanner设计了引入指代处理的模块。其实现步骤可以表示如下：

1. 遍历所有的组块，找到所有的名词，并生成一个名词序列。
2. 对名词序列中的每一个名词，通过查询知识库中与其相关的RelEx关系，选定能在替换该名词时所要使用的代词的根，这里的根指”he”,”she”,”it”,”they”等。例如，“Mary”通常是表示女名，那么这里就指定能替换”Mary“的代词的根”she“。
3. 遍历每个名词，判断它是否可以被替换成代词：

* 如果该名词从来没有被提及过，或者在很久前被提及过（超过n个组块的范围，n默认为3），那么该名词不能被替换成代词。
* 如果该名词在同一个组块里被InheritanceLink修饰，那么该名词不能被替换成代词。（英文中形容词后面的名词不能被替换成代词）
* 检查该名词在名词序列中的前3个词，如果其中一个已经被替换成代词了，那么为了降低表达的歧义，这里也认为该名词不能被替换成代词。

1. 根据上面的判断结果，对可以被替换成代词的名词进行如下代词替换操作：

* 如果名词是主语，将代词保持主格形式，（即”I”,”he”,”they”等）
* 如果名词在被赋值为“所有格”的EvaluationLink中，那么将其改为所有格形式（“my”“his”“its”等）
* 如果名词是宾语或者间接宾语
* 如果它和主语是相同名词，那么根据情况替换成“myself”，“himself”，“themselves”等。
* 如果它和主语不同，那么根据情况替换成”me”,”him”,”them”等

1. 如果一个名词无法被换成代词，将使用下列算法来进一步考虑它是否可以被相关的名词替代（例如“松树”可以用“树”取代）：

* 通过InheritanceLink查找与该名词节点有继承关系的名词节点
* 根据下面公式对这些有继承关系的名词进行打分：( )∗( InheritanceLink Strength)∗( InheritanceLink Confidence)
* 选择得分最高的名词去替换（如果最高得分的名词不止一个，那么随机选择一个）

### 存在问题和改进方向

我们这里讨论的Microplanner虽然能处理一般的现象，并且能和后面章节要讨论的对话管理以及表层生成等模块结合起来用于简单的对话系统。但还有很多方面需要改进。一般来说，微观规划过程还包含选词处理，但由于我们目前使用的知识库还比较小，可以直接通过InheritanceLink找到相关词，所以这一模块暂时被搁浅，我们会在下一个版本中实现选词模块以及加入更多话语类型的处理模块。除此之外，我们还可以在微观规划过程中引入简单的逻辑理论，进一步改进指代的引入。

## 表层生成器SuReal

表层生成是将经微观规划后的内容描述映射至由文字、标点符号和结构注解信息组成的表层文本的过程。在第1.2节中提到，表层生成的算法有很多种，而且由于本身“表层生成”的概念没有明确的定义，不同的表层生成系统的输入都不同，很难有可比性。本文从人脑或类人脑系统的生成语言的思路出发，结合机器具备大型计算能力的优点，设计并在OpenCog中的知识库上实现了基于超图近似匹配的表层生成器（Surface Realizer，以下简称SuReal），使其能将微观规划后的超图转换成英文句子。本节将会具体阐述其工作原理和实现方法。

### SuReal算法

概括地说，SuReal的工作可以看成是第5中描述的自然语言理解系统Link Parser,RelEx,RelEx2Logic的逆过程，但是，实现这样的逆转换并没那么简单，因为在将英文句子转成逻辑形式的过程中抽象了很多词法或语法关系等重要语言表层现象。而且在我们的自然语言系统中创建的一系列规则也只是同态而并不同构，无法通过简单地反向使用同样的规则集来实现这个逆转过程。

基于这些考虑，我们提出了一个全新的表层生成方法，它以基于超图的逻辑表达式作为输入，但不是直接在语义层面上将此逻辑表达式与知识库中的其他逻辑表达式进行匹配，而是将句法分析结果也转换成超图形式引入到知识库中，然后在句法规则的指导下进行基于超图的匹配。SuReal的实现原理操作流程如下：

* 使用第5章中的自然语言理解系统分析一系列英文句子，然后将分析后得到的语义结构以及中间步骤得到的句法结构形成配对的二元组，以（语言表达式，句法表达式+逻辑表达式）的形式存入到知识库Atomspace中。其中逻辑表达式是指RelEx2Logic产生的使用超图表示的逻辑形式，句法表达式是链语法分析产生的结果并存入到Atomspace中的超图形式；语言表达式即以超图中的SentenceNode节点表示的英文句子。句法表达式在Atomspace中的超图表示如下：

;链语法分析结构中”the”和“cat”之间的链接的超图表示

(EvaluationLink (stv 1.0 1.0)

(LgLinkInstanceNode "Ds\*\*c@443") (ListLink

(WordInstanceNode "the@da6")

(WordInstanceNode "cat@1a6")

)

)

;链语法分析结构中链 Ds\*\*c 的超图表示

(LgLinkInstanceLink (LgLinkInstanceNode "Ds\*\*c@443") (LgConnector

(LgConnectorNode "D") (LgConnDirNode "+")

)

(LgConnector

(LgConnectorNode "Ds\*\*c") (LgConnDirNode "-")

)

)

* SuReal的输入需要用SetLink连接用于生成句子的逻辑表达式，这里假设输入为SetLink S，那么可通过以下操作实现SuReal(S)：

1. 对给定的SetLink S中的某个节点元素，在链语法词典中查找对应的链语法链接要求，并以DNF(Disjunctive Normal Form)的形式存入Atomspace中
2. 调用OpenCog中的模式匹配器（Pattern Matcher），将S中的每个节点携带相应链语法链接要求的词节点当成变量，与预先存入知识库中的二元组中的逻辑表达式和句法表达式进行匹配。
3. 当模式匹配器在知识库中搜索到对应的匹配后，检查其中的词节点携带的链语法链接要求是否与S中对应节点携带的链语法链接要求DNF是否吻合，如果含有相似的链接要求，则将匹配到的超图中的词节点替换成S中对应的词节点。
4. 将匹配结果在二元组中对应的语言表达式中的相应词替换成新的词节点对应的词后，生成句子。
5. 对上一步得到的所有句子进行排序后输出最终结果。排序中考虑的因素有语言模型、已经原有句子中被替换掉的词的个数等。

### 综观SuReal及其存在的问题和改进方向

广义上来说，本文提出的SuReal的算法可以看成是一个超图的改写过程，这个改写过程和前文中阐述的自然语言理解流程（即Link Parser,RelEx and RelEx2Logic）相反。然而，在具体实现上，这一过程并非容易，因为自然语言理解流程中的语法语义规则都是同态而非同构。对于表层生成来说，任何一个知识库中的节点结构都可能由很多不同的链语法结构产生，因为对于同一种说法或句子，会有很多不同的表达方式，从而产生很多不同的链语法结构，但是并非所有这些链语法结构之间都需要相配。因此，SuReal将表层生成过程看成是一个约束满足问题（constraint satisfaction problem），而无论从句法歧义角度，还是从人类对句子是否自然的主观看法角度来看，该问题通常都有很多解答。上述的算法即是通过启发式方法去实现和解决这一问题。

更精确地说，假设我们有原子集合A={Ai}，使R={R j}表示所有的超图改写规则R j的集合；其中R j至少能将一个链语法分析子树与{Ai}中某个非空原子集合匹配。假设Ri∈R表示改写规则集合中能产生包含Ai在内的匹配的规则的子集，因此我们可以将其表示如下Ri={Ri},其中R=∪i Ri。又假设mg(r)表示改写规则r能匹配超图g中的某个子图。那么根据以上这些假设，表层生成的问题可看成是，对于表示需要表达内容的原子集合A，查找与其相匹配的并且满足下列条件的语法分析对应的超图g：

*k*

* 必须是符合链语法的语法规则
* 必须满足“可表达性”的条件，即

∧ ∨ *mg*(*Ri* ),

*k*

*i k*

* 必须满足系统规定的“审美条件”，目前我们系统将该条件定义为：删除任意一个词后，该结构依然满足上述两个条件。

上述过程完成了将表示想表达的内容的原子集合A中抽取了能通过一个合法自然语言句子来表达的子集，当找到这样的链语法分析结构之后，生成相应的自然语言句子就很显然。

对于上述的SuReal使用的方法，在多数情况下，该约束满足问题都能找到很多解答，而如何从这些解答中找出最优解的方法也很多，本文采用的方法是：通过在大型语料中训练出来的语言模型来对生成的句子进行打分，从而选择得分最高的解答。

那么目前SuReal存在的问题有：

* 受限于RelEx2Logic的输出。由于匹配所使用的三元组知识库是通过我们的自然语言理解系统产生的，那么其中的逻辑表达式中的所有Atoms类型只能来源于RelEx2Logic中定义的，因此如果输入中含有RelEx2Logic中未使用过的Atom类型，那么将无法生成句子。
* 数据驱动，如果知识库中没有与想表达内容相关的句子，那么系统将会随机生成包含相关单词的句子，从而无法正确表达。
* 系统目前使用的查找匹配过程比较繁琐，并未将强化学习使用到匹配中，使匹配过程更智能简洁。

可以改进方向包括：

* 使用强化学习来实现更智能更灵活的超图匹配。
* 改进RelEx2Logic使其能提供更丰富的逻辑关系，从而使通过我们的自然语言理解系统分析并存入知识库的查询库更丰富。

### 实验结果示例及简要分析

正如前面提到，不同的自然语言生成系统所使用的输入形式都不一样，有词语层次的输入，也有句法和语义层次的输入，导致研究的方向不一样，算法也千差万别，因此目前并没有权威的评估标准。我们这里只给出本系统的一些实验结果并简要分析。

首先给出一个句子集如下：

Play a song by Weird Al Yankovic.

Play another song by Weird Al.

Play something by The Cure.

Who wrote 'Blue Monday'?

What is the best song by New Order?

When did 'Thriller' come out?

I want to hear some 60's soul music.

Can you play something new for me?

He bought a guitar in the store.

Madonna sang the song called 'Vogue'.

然后运行我们的自然语言处理系统，将得到的相应的（语言表达式，逻辑表达 式）二元组导入 OpenCog 的知识库 Atomspace 中。运行 SuReal 可以得到如下实验 结果：

输入：

(SetLink

(EvaluationLink

(PredicateNode "ate")

(ListLink

(ConceptNode "John")

(ConceptNode "pig")

)

)

)

输出：

(John ate a pig.)

(John ate the pig.)

不难看出，第一个输出(Jonh ate a pig.)是匹配导入的句子集合中“He bought a guitar in the store.”对应的超图的子图成功后得到的结果；第二个输出(John ate the pig.)是匹配其中“Madonna sang the song called’Vogue’.”对应的超图的子图成功后得到的结果。

输入：

(SetLink

(EvaluationLink

(PredicateNode "sang")

(ListLink (VariableNode "$ABC"))

)

)

输出：

(who sang ['] Blue Monday ['] ?)

该输入中含有一个变量，说明是由Microplanner规划的疑问句形式，而且根据输入的逻辑表达式可以看出，该变量充当主语的角色。因此可以通过匹配“Who wrote’Blue Monday’?”后得到相应的结果。

## 本章小节

本章针对自然语言生成，设计并实现了一个新的自然语言生成系统，能将基于超图表示的逻辑形式转换成英语句子形式。由于使用独特的知识表示系统，该设计完全是本文特有的构想。当然，该语言生成系统也有一定的局限性，比如Microplanner目前只能处理一定的话语类型，SuReal无法生成知识库中不存在的表达。但我们的实验结果表明，它们已经能完全运用在简单的句子生成上，有时候甚至也能处理稍微复杂一点的情况。

# 

第七章 基于超图表示的语言逻辑推理的设计与实现

基于自然语言的逻辑推理在很多应用领域都起到了非常重要的作用，其中也包 括本文第[8](#_bookmark106)章将介绍的认知对话系统。针对第[2](#_bookmark24)章阐述的基于超图的知识表示以及逻 辑推理系统等的理论知识，并在第[5](#_bookmark66)章介绍的自然语言理解系统的基础上，我们进 一步设计并实现了一个基于超图表示的语言逻辑推理系统。本章将通过几个典型例 子来介绍该语言逻辑推理系统的设计思路和实现步骤，也就是将自然语言通过自然 语言理解系统转换成超图的表示形式，并使用概率逻辑网 PLN 来进行一些常识推 理的过程。



## 有关比较级的推理

首先，紧接着上一章中对比较级的句法分析的讨论，我们这里通过几个例子来 展示，在我们的自然语言系统输出的有关比较级的逻辑表达式上，PLN 如何实现相 关推理。为了更好地显示有效的推理过程，我们在下面的例子中过滤了和比较级推 理无关的 RelEx 关系和逻辑表达。

假设有：

* Bob likes Hendrix more than the Beatles.
* Bob is American.
* Menudo is liked less by Americans than the Beatles.

那么我们可以推出：Bob likes Hendrix more than Menudo.

对于第一个句子，通过执行 RelEx 和 RelEx2Logic，可以得到：

\_subj(like, Bob)

\_obj(like, Hendrix)

than(Hendrix, Beatles)

\_comparative(like, Hendrix)

==>

TruthValueGreaterThanLink

EvaluationLink like Bob Hendrix

EvaluationLink like Bob Beatles

类似地，第二个句子：

\_subj(like, Americans)

\_obj(like, Menudo) than(Beatles, Menudo)

\_comparative(like, Beatles)

==>

TruthValueGreaterThanLink

EvaluationLink like Americans Beatles

EvaluationLink like Americans Menudo

从这些句子中得到的逻辑表达是非常透明的。简单地根据“TruthValue Greater Than” 关系是可传播，以及“Bob is American”，PLN很容易推理得到结论：

TruthValueGreaterThanLink

EvaluationLink like Bob Hendrix

EvaluationLink like Bob Menudo

那么现在所处理的知识是逻辑形式而非句法形式，因此可以调用知识库中的相关知识加入到推理过程中。例如，假设我们还知道“Bob likes Sinatra more than Menudo”，可表示如下：

TruthValueGreaterThanLink

EvaluationLink like Bob Hendrix

EvaluationLink like Bob Menudo

根据 PLN 中的回溯推理规则 *((A → C) ∧ (B → C) ⇒ (A → C))*，可以得到：

SimilarityLink

Hendrix

Sinatra

## 基于自然语言的三段论推理

为了测试RelEx2Logic的输出用于逻辑推理的有效性，我们使用了PLN来执行基于自然语言输入的三段论的推理。接下来，我们给出一个基于自然语言的三段论推理的例子，和详细的推理步骤。

假设我们需要完成的推理如下：

Socrates is a man.（苏格拉底是人）

Men breath air.（人呼吸空气）

|-

Socrates breath air.（苏格拉底呼吸空气）

对于此例，推理前提在 Atomspace 中可表示如下：

Concepts

(ConceptNode "Socrates@f25") ; [217]

(ConceptNode "Socrates" (stv .1 1.0)) ; [218]

(ConceptNode "man@80d" (stv .1 1.0)) ; [220]

(ConceptNode "men@83d" (stv .1 1.0)) ; [290]

(ConceptNode "man" (stv .1 1.0)) ; [221]

(ConceptNode "air@e3f") ; [292]

(ConceptNode "air") ; [293] (ConceptNode "present") ; [227]满足 breathe(x,y的多元组可表示如：) (ListLink (stv 1.000000 .000000)

(ConceptNode "men@83d") ; [290]

(ConceptNode "air@e3f") ; [292]

) ; [295]

breathe(x,y)

(EvaluationLink(stv 1.000000 .000000)

(PredicateNode "breathe@218") ; [287] (ListLink (stv 1.000000 0.000000)

(ConceptNode "men@83d") ; [290] (ConceptNode "air@e3f") ; [292]

) ; [295]

) ; [296]

"Socrates IS-A man（“苏格拉底是人”）可表示如下：" (InheritanceLink (stv 1.000000 1.000000)

(ConceptNode "Socrates@f25") ; [217] (ConceptNode "man@80d") ; [220]

) ; [223]

具体实例继承于泛化概念：

(InheritanceLink (stv 1.000000 0.990000) (ConceptNode "men@83d") ; [290] (ConceptNode "man") ; [221]

) ; [291]

(InheritanceLink (stv 1.000000 1.000000) (ConceptNode "Socrates@f25") ; [217] (ConceptNode "Socrates") ; [218]

) ; [219]

(InheritanceLink (stv 1.000000 0.990000) (ConceptNode "man@80d") ; [220] (ConceptNode "man") ; [221]

) ; [222]

下图显示了 PLN 的推理路径以及每一步中使用的推理规则：

1. 成员变量泛化规则（）GeneralEvaluationToMemberRule输入：

(EvaluationLink (stv 1.000000 1.000000) (PredicateNode "breathe@7f5") ; [350] (ListLink (stv 1.000000 0.000000)

(ConceptNode "men@a29") ; [353] (ConceptNode "air@9bf") ; [355]

) ; [358]

) ; [359]输出：以及

(中其他变量被泛化后的个其他变体）ListLink3 (MemberLink (stv 1.000000 1.000000)

(ConceptNode "men@a29") ; [353] (SatisfyingSetLink (stv 1.000000 1.000000)

(VariableNode "$X0") ; [441] (EvaluationLink (stv 1.000000 0.000000)

(PredicateNode "breathe@7f5") ; [350] (ListLink (stv 1.000000 0.000000)

(VariableNode "$X0") ; [441] (ConceptNode "air@9bf") ; [355]

) ; [443]

) ; [444]

) ; [445]

) ; [446]

1. 成员继承规则（MemberToInheritance ）Rule输入：上一步的输出输出：

(InheritanceLink (stv 1.000000 1.000000) (ConceptNode "men@a29") ; [353] (SatisfyingSetLink (stv 1.000000 1.000000)

(VariableNode "$X0") ; [441] (EvaluationLink (stv 1.000000 0.000000)

(PredicateNode "breathe@7f5") ; [350] (ListLink (stv 1.000000 0.000000)

(VariableNode "$X0") ; [441] (ConceptNode "air@9bf") ; [355]

) ; [443]

) ; [444]

) ; [445]

) ; [6107]

1. 溯因推理规则（）AbductionRule输入：

(InheritanceLink (stv 1.000000 0.990000) (ConceptNode "men@a29") ; [353] (ConceptNode "man") ; [284]

) ; [354]

(InheritanceLink (stv 1.000000 0.990000) (ConceptNode "man@fbc") ; [283] (ConceptNode "man") ; [284]

) ; [285]输出：

(InheritanceLink (stv 1.000000 0.988901) (ConceptNode "man@fbc") ; [283] (ConceptNode "men@a29") ; [353]

) ; [609]

4) 演绎推理规则（）DeductionRule输入：上一条输出以及下面的原子集合

(InheritanceLink (stv 1.000000 1.000000) (ConceptNode "Socrates@46e") ; [280] (ConceptNode "man@fbc") ; [283]

) ; [286]输出：

(InheritanceLink (stv 1.000000 0.991755) (ConceptNode "Socrates@46e") ; [280] (ConceptNode "men@a29") ; [353]

) ; [777]

5) 演绎推理规则（）DeductionRule输入：步骤）和步骤）的输出

24输出：

(InheritanceLink (stv 1.000000 0.986333) (ConceptNode "Socrates@46e") ; [280] (SatisfyingSetLink (stv 1.000000 0.000000)

(VariableNode "$X1") ; [442] (EvaluationLink (stv 1.000000 1.000000)

(PredicateNode "breathe@7f5") ; [350] (ListLink (stv 1.000000 0.000000)

(VariableNode "$X1") ; [442]

(ConceptNode "air@9bf") ; [355]

) ; [922]

) ; [923]

) ; [8605]

) ; [12317]

* 1. 成员变量继承规则（）InheritanceToMemberRule输入：上一条规则的输出输出：

(MemberLink (stv 1.000000 0.989841) (ConceptNode "Socrates@46e") ; [217] (SatisfyingSetLink (stv 1.000000 1.000000)

(VariableNode "$X0") ; [385] (EvaluationLink (stv 1.000000 0.000000)

(PredicateNode "breathe@7f5b") ; [350] (ListLink (stv 1.000000 0.000000)

(VariableNode "$X1") ; [442] (ConceptNode "air@9bf") ; [355]

) ; [922]

) ; [388]

) ; [389]

) ; [6375]

7) 成员变量赋值规则（）MemberToEvaluationRule输入：上一条规则的输出最终推理结论：

(EvaluationLink (stv 1.000000 0.989841) (PredicateNode "breathe@7f5") ; [350] (ListLink (stv 1.000000 0.000000)

(ConceptNode "Socrates@46e") ; [217] (ConceptNode "air@9bf") ; [355]

) ; [922]

) ; [388]

需要说明的是，上述提供的从前提到结论的推理路径中，只显示了在推理过程中提供有用信息的推理步骤。在推理过程中，PLN的后向推理链尝试并放弃大量其他的可能推理步骤。

## 本章小结

本章介绍了几个使用自然语言理解系统的输出在PLN上的推理实验。实验表明，我们的自然语言理解系统输出的逻辑形式能和逻辑推理引擎结合，并能进行简单的常识推理。但是目前的系统还无法完成较复杂的推理。我们下一步将继续改进RelEx2Logic系统，使其能更好地为复杂的句子提供更准确的逻辑表达，并改善PLN的控制机制，使其能进行更复杂的推理。

# 

第八章 基于概率逻辑推理的问答系统的设计和实现

本文的智能对话系统的预期目标主要是实现面向游戏角色和人型机器人的对话系统，但同样的研究思路也完全适用于基于文本的对话系统，例如智能对话搜索接口。我们分阶段来实现这样的目标，首先以一些相对小二简单的智能对话系统为起点，通过不断改进和完善，以及结合系统本身的自学习和人只能里，最终实现一个在一定程度上接近人类智能水平的对话系统。本章就根据第4章中提出的智能对话系统CogDial的概念模型，并结合第5章的自然语言理解技术以及第6章自然语言生成技术的研究成果，设计并实现了几个基于概率逻辑推理的问答系统。除此之外，本章还结合我们研究中的发现和遇到的难题对深层语义解析的挑战进行了讨论，也给未来的研究提供了一个明确的方向。

需要说明的是，目前该系统只能处理英文对话，所以接下来介绍系统的设计和实现时所举的对话例子基本是按照英文习惯，可能不太适合中文的语言习惯，但是该系统所采用的技术和理论基础都是完全适合中文的。



## 基于超图模糊匹配的问答系统

为了实现我们的智能对话系统的框架以及设计思路，我们构建了一个基于超图模糊匹配的问答系统的原型，该问答系统使用前文提到的自然语言理解技术将自然语言转换成以超图表示的语义逻辑形式，利用基于超图匹配的动态编程算法，从而在问答语料库中找到最适合问题的答案来响应用户查询。

本节首先讨论该问答系统框架中使用的超图匹配算法，然后再讨论基于超图模糊匹配的问答系统的设计和实现。

### 基于超图的模式匹配器

基于超图的模式匹配器是一个查询引擎或变数配对者，主要功用是在我们的超图知识库Atomspace中寻找特定的模式，或是一些Atom相关排列或“模板“。在轮入一个等定的Atom排列（即一个超图）后，模式匹配器会从Atomspace中寻找所有合乎条件的超图。同时一些“空白位置“，即“变数“在一个超图的位置，亦可存在于该输入的超图当中，而模式匹配器则会“填补“这些“空白位置“。例如：“John threw a ).“亦可以被判断为“John threw a ball.“，前提是在查询前该句子要存在于Atomspace里，而在这个例子中“ball“就是被配对的一个答案。输入的超图中可以在不同的地方拥有多过一个的“空白位置“，同时亦可以有多过一个的答案。在这过程中，BindLink提供了一个便利的API去达到这目的，接下来会有更多的技术层面说明。

模式匹配器是一个变数配对者，是在传统计算机科学中“统一“的概念，因为这是它的功能之一。它同时亦是一个查询引擎，因为这个变数配对过程某情度上是等同于用SQL在关联数据库中进行查询。其中主要的分别在于在我们这里的概念是以超图的形式表示，所以它亦可以被形容为一个超图查询语言HQL（Hypergraph Query Language）。但这些都只是以不同的名称去表示同一个程序。

这个模式匹配器是结构精密应用的一个重要组件，同时也是建立逻辑推理系统中前向和后向链接推理的一个重要基础。它拥有C++和Scheme的接口以供不同的应用。

模式匹配器由数个不同的组件连接在一起，组成一个基础前向链接或超图重写的工具。在其核心是一个能比较和配对不同超图以及其变数的组件，而这个组件跟另一个“找寻器“一同使用便能寻找整个Atomspace中的合乎条件的超图。最后一个组件是一个“编写器“，主要功用是当有一个配对成功产生后建立一个或以上在Implic-

ationLink后半部列明的超图。

这个模式匹配器接受有指定“规则“的BindLinks，再把这些“规则“应用到Atomspace里。每一个“规则“是一个ImplicationLink，以“if P then Q“的形式表达，当中P和Q分别代表不同的超图。如果P被实现了，那么便能得到Q。Q可以是任何种类的超图，但如果Q是一个ExecutionLink，这便意味著当P被实现了，系统便将会实行其他程序，这在整体来说可以是一个十分强大的功能，因为在一般情况下很多程序都可以以ExecutionLink来执行。

模式匹配器不会更改任何超图中的真值（Truth Value）或关注值（Attention Value），有需要时使用者亦可以自行编写和执行相关的程序。

在默认的回调函数中模式匹配器会搜索整个Atomspace，因此亦有需要编写特定的回调函数以限制其搜索范围，例如只寻找和接受拥有某程度短期重要性的Atom，以只获取相对重要的资讯。

以下是一个使用模式匹配器的示例：

(define find-animals (BindLink

;; The variable to be bound (VariableNode "$var") (ImplicationLink

;; The pattern to be searched for (InheritanceLink

(VariableNode "$var") (ConceptNode "animal")

)

;; The value to be returned. (VariableNode "$var")

)

)

)

执行时，只需在我们系统的Scheme终端输入以下指令便能执行以上的模式匹配器并找出在Atomspace中所有继承“animal“的概念：

(cog-bind find-animals)

### 基于超图模糊匹配的问答系统

本小节我们将阐述这个基于超图的模糊匹配的启发式的问答系统的设计思路和实现方法，该系统主要针对信息查询。其基本算法可简单概括如下：给定一个查询Q，通过我们的自然语言理解流水线将其转换成Atomspace里的超图形式，然后在知识库Atomspace里找出与其相似的表示陈述句的超图。这里我们做了一个合理的猜测，认为这些相似的超图中的其中一个包含了Q的答案。

如果给定一个很大的超图H，在H的所有子图中找到与目标超图Q部分匹配的子超图是一个非常困难的计算问题。我们使用了启发式在一定程度上解决了计算难度，虽然这不能保证找到问题的正确答案，但实验发现该方法通常能找到一个很好的答案。我们的启发式涉及下面两个阶段：

* 第一阶段：使用一个基于超图的模糊匹配器Pattern Matcher，搜索能正确回答查询Q的答案。在此搜索过程中，保存所有与查询Q近似匹配的子超图。
* 第二阶段：对第一阶段中的所有部分匹配的子超图，使用基于超图匹配的动态规划来计算它们的匹配程度，然后根据匹配程度进行排序。

基于超图匹配的动态规划是对[119]中的算法一个新的改进。正如基于动态规划的字符串匹配，获得高性能的一个重要因素是从源到目标的路径转换过程中合理地分配具体的操作成本。我们对此使用的启发式如下：修改（可以是添加、删除或替换）一个稀有词对应的节点比修改一个常用词对应的节点的成本高；修改一个稀有词节点对应的链比修改一个常用词节点对应的链的成本高。这里的“稀有程度”通过词频或者该词所关联的WordNode,ConceptNode或者PredicateNode的真值来衡量。

假设所需的查询是“Who bought a glockenspiel at the store?”(谁在商店买了一个钟琴？)，那么如果将“glockenspiel”替换成其他词的时候，那其成本就比将“store”替换成其他词要高，因为“glockenspiel”比“store”更稀有。因此“Bob bought a glockenspiel with his friend.”（Bob和他朋友一起买了钟琴。）和查询的匹配程度要高于“Jim bought a thimble at the store.”（Jim在商店买了一个针箍。）。

这种基于频率的启发式相当于实例化OpenCog里经常使用的一个一般原则：信息含量往往通过概率化的惊讶度来衡量。发现句子里的“glockenspiel”比发现“store”的惊讶度更高，因此，我们断定，在此查询中，“glockenspiel”含有的信息量更大，所以当修改“glockenspiel”的时候会被分配很高的成分，因为做这样的修改就相当于删除了查询中的较多的信息。这样的信息论原则也给OpenCog中的Pattern Mining[120]奠定了基础,Pattern Mining在OpenCog的动机机制中充当了重要的角色，可以用来满足我们前面提到的OpenPsi里的“新颖性”（Novelty）的目标需求。一个更智能更复杂的匹配方法可以根据总体惊讶值来惩罚修改序列，但目前的做法依然是只针对查询的各个部分的惊讶度单独作为指标，待系统逐步改善后，会考虑实现更复杂的匹配算法。为了简化操作，在做查询匹配的过程中，我们忽略了表示查询的超图中很多不重要的Atoms，仅表明语义关系的核心部分用于匹配。对于例句“Who bought a glockenspiel at the store?”，仅有下面这些Atoms参与超图匹配：

((ReferenceLink

(InterpretationNode "sentence@ae4\_parse\_0\_interpretation\_$X") (SetLink

(ImplicationLink (stv 0.99000001 0.99000001)

(PredicateNode "bought@530" (stv 0.001 0.99000001))

(PredicateNode "buy" (ptv 0.001 0.99000001 1))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(ConceptNode "glockenspiel@bf4" (stv 0.001 0.99000001))

(ConceptNode "glockenspiel" (ptv 0.001 0.99000001 1))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(ConceptNode "store@ae4" (stv 0.001 0.99000001))

(ConceptNode "store" (ptv 0.001 0.99000001 1))

)

(EvaluationLink (stv 0.99000001 0.99000001)

(PredicateNode "bought@530" (stv 0.001 0.99000001))

(ListLink (stv 0.99000001 0.99000001)

(VariableNode "$rIr" (stv 0.001 0.99000001))

(ConceptNode "glockenspiel@bf4" (stv 0.001 0.99000001))

(ConceptNode "store@ae4" (stv 0.001 0.99000001))

)

)

(InheritanceLink (stv 0.99000001 0.99000001)

(ConceptNode "at@d68" (stv 0.001 0.99000001))

(ConceptNode "at" (ptv 0.001 0.99000001 1))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(SatisfyingSetLink (stv 0.99000001 0.99000001)

(PredicateNode "bought@530" (stv 0.001 0.99000001))

)

(ConceptNode "at@d68" (stv 0.001 0.99000001))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(PredicateNode "bought@530" (stv 0.001 0.99000001))

(ConceptNode "past" (stv 0.001 0.99000001))

)

(EvaluationLink (stv 0.99000001 0.99000001)

(PredicateNode "definite" (stv 0.001 0.99000001))

(ListLink (stv 0.99000001 0.99000001)

(ConceptNode "store@ae4" (stv 0.001 0.99000001))

)

)

(ImplicationLink (stv 0.99000001 0.99000001)

(PredicateNode "at@d68" (stv 0.001 0.99000001))

(PredicateNode "at" (ptv 0.001 0.99000001 1))

)

(EvaluationLink (stv 0.99000001 0.99000001)

(PredicateNode "at@d68" (stv 0.001 0.99000001))

(ListLink (stv 0.99000001 0.99000001)

(ConceptNode "store@ae4" (stv 0.001 0.99000001))

)

)

(InheritanceLink (stv 0.001 0.99000001)

(InterpretationNode "sentence@ae4\_parse\_0\_interpretation\_$X") (ConceptNode "InterrogativeSpeechAct" (stv 0.001 0.99000001))

)

)

)

)

类似地，我们使用简化过的表示“Bob mauled a glockenspiel with his friend.”的超图：

((ReferenceLink

(InterpretationNode "sentence@477\_parse\_0\_interpretation\_$X") (SetLink

(ImplicationLink (stv 0.99000001 0.99000001)

(PredicateNode "mauled@1a0" (stv 0.001 0.99000001))

(PredicateNode "maul" (ptv 0.001 0.99000001 1))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(ConceptNode "Bob@1c3" (stv 0.001 0.99000001))

(ConceptNode "Bob" (ptv 0.001 0.99000001 1))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(ConceptNode "glockenspiel@b12" (stv 0.001 0.99000001))

(ConceptNode "glockenspiel" (ptv 0.001 0.99000001 1))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(ConceptNode "friend@e78" (stv 0.001 0.99000001))

(ConceptNode "friend" (ptv 0.001 0.99000001 1))

)

(EvaluationLink (stv 0.99000001 0.99000001)

(PredicateNode "mauled@1a0" (stv 0.001 0.99000001))

(ListLink (stv 0.99000001 0.99000001)

(ConceptNode "Bob@1c3" (stv 0.001 0.99000001))

(ConceptNode "glockenspiel@b12" (stv 0.001 0.99000001))

(ConceptNode "friend@e78" (stv 0.001 0.99000001))

)

)

(InheritanceLink (stv 0.99000001 0.99000001)

(ConceptNode "with@8ef" (stv 0.001 0.99000001))

(ConceptNode "with" (ptv 0.001 0.99000001 1))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(SatisfyingSetLink (stv 0.99000001 0.99000001)

(PredicateNode "mauled@1a0" (stv 0.001 0.99000001))

)

(ConceptNode "with@8ef" (stv 0.001 0.99000001))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(PredicateNode "mauled@1a0" (stv 0.001 0.99000001))

(ConceptNode "past" (stv 0.001 0.99000001))

)

(InheritanceLink (stv 0.001 0.99000001)

(InterpretationNode "sentence@477\_parse\_0\_interpretation\_$X") (ConceptNode "DeclarativeSpeechAct" (stv 0.001 0.99000001))

)

(EvaluationLink (stv 0.99000001 0.99000001)

(PredicateNode "definite" (stv 0.001 0.99000001))

(ListLink (stv 0.99000001 0.99000001)

(ConceptNode "friend@e78" (stv 0.001 0.99000001))

)

)

(InheritanceLink (stv 0.99000001 0.99000001)

(ConceptNode "his@301" (stv 0.001 0.99000001))

(ConceptNode "his" (ptv 0.001 0.99000001 1))

)

(EvaluationLink (stv 0.99000001 0.99000001)

(PredicateNode "possession" (stv 0.001 0.99000001))

(ListLink (stv 0.99000001 0.99000001)

(ConceptNode "friend@e78" (stv 0.001 0.99000001))

(ConceptNode "his@301" (stv 0.001 0.99000001))

)

)

(ImplicationLink (stv 0.99000001 0.99000001)

(PredicateNode "with@8ef" (stv 0.001 0.99000001))

(PredicateNode "with" (ptv 0.001 0.99000001 1))

)

(EvaluationLink (stv 0.99000001 0.99000001)

(PredicateNode "with@8ef" (stv 0.001 0.99000001))

(ListLink (stv 0.99000001 0.99000001)

(ConceptNode "friend@e78" (stv 0.001 0.99000001))

)

)

(InheritanceLink (stv 0.99000001 0.99000001)

(SpecificEntityNode "Bob@1c3" (stv 0.001 0.99000001))

(ConceptNode "male" (stv 0.001 0.99000001))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(SpecificEntityNode "Bob@1c3" (stv 0.001 0.99000001))

(ConceptNode "Bob" (ptv 0.001 0.99000001 1))

)

(EvaluationLink (stv 0.99000001 0.99000001)

(PredicateNode "definite" (stv 0.001 0.99000001))

(ListLink (stv 0.99000001 0.99000001)

(ConceptNode "Bob@1c3" (stv 0.001 0.99000001))

)

)

)

)

)

和表示”Jim bought a thimble at the store.”的超图

((ReferenceLink

(InterpretationNode "sentence@d92\_parse\_0\_interpretation\_$X") (SetLink

(ImplicationLink (stv 0.99000001 0.99000001)

(PredicateNode "bought@7e4" (stv 0.001 0.99000001))

(PredicateNode "buy" (ptv 0.001 0.99000001 1))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(ConceptNode "Jill@bc3" (stv 0.001 0.99000001))

(ConceptNode "Jill" (ptv 0.001 0.99000001 1))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(ConceptNode "thimble@8a3" (stv 0.001 0.99000001))

(ConceptNode "thimble" (ptv 0.001 0.99000001 1))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(ConceptNode "store@ef6" (stv 0.001 0.99000001))

(ConceptNode "store" (ptv 0.001 0.99000001 1))

)

(EvaluationLink (stv 0.99000001 0.99000001)

(PredicateNode "bought@7e4" (stv 0.001 0.99000001))

(ListLink (stv 0.99000001 0.99000001)

(ConceptNode "Jill@bc3" (stv 0.001 0.99000001))

(ConceptNode "thimble@8a3" (stv 0.001 0.99000001))

(ConceptNode "store@ef6" (stv 0.001 0.99000001))

)

)

(InheritanceLink (stv 0.99000001 0.99000001)

(ConceptNode "at@50d" (stv 0.001 0.99000001))

(ConceptNode "at" (ptv 0.001 0.99000001 1))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(SatisfyingSetLink (stv 0.99000001 0.99000001)

(PredicateNode "bought@7e4" (stv 0.001 0.99000001))

)

(ConceptNode "at@50d" (stv 0.001 0.99000001))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(PredicateNode "bought@7e4" (stv 0.001 0.99000001))

(ConceptNode "past" (stv 0.001 0.99000001))

)

(InheritanceLink (stv 0.001 0.99000001)

(InterpretationNode "sentence@d92\_parse\_0\_interpretation\_$X") (ConceptNode "DeclarativeSpeechAct" (stv 0.001 0.99000001))

)

(EvaluationLink (stv 0.99000001 0.99000001)

(PredicateNode "definite" (stv 0.001 0.99000001))

(ListLink (stv 0.99000001 0.99000001)

(ConceptNode "store@ef6" (stv 0.001 0.99000001))

)

)

(ImplicationLink (stv 0.99000001 0.99000001)

(PredicateNode "at@50d" (stv 0.001 0.99000001))

(PredicateNode "at" (ptv 0.001 0.99000001 1))

)

(EvaluationLink (stv 0.99000001 0.99000001)

(PredicateNode "at@50d" (stv 0.001 0.99000001))

(ListLink (stv 0.99000001 0.99000001)

(ConceptNode "store@ef6" (stv 0.001 0.99000001))

)

)

(InheritanceLink (stv 0.99000001 0.99000001)

(SpecificEntityNode "Jill@bc3" (stv 0.001 0.99000001))

(ConceptNode "female" (stv 0.001 0.99000001))

)

(InheritanceLink (stv 0.99000001 0.99000001)

(SpecificEntityNode "Jill@bc3" (stv 0.001 0.99000001))

(ConceptNode "Jill" (ptv 0.001 0.99000001 1))

)

(EvaluationLink (stv 0.99000001 0.99000001)

(PredicateNode "definite" (stv 0.001 0.99000001))

(ListLink (stv 0.99000001 0.99000001)

(ConceptNode "Jill@bc3" (stv 0.001 0.99000001))

)

)

)

)

)

我们的下一步研究会将逻辑推理也加入成本计算的过程，从而进一步改进上述针对查询处理的超图匹配。假如知识库中代表下面这个句子的超图表示“Jim bought a musical instrument at the store.”(Jim在商店买了一个乐器。)，那么如果该查询系统能做出以下推理：

InheritanceLink

ConceptNode "glockenspiel"

ConceptNode "musical instrument"

得到“钟琴”是一种“乐器”，那么会对“钟琴”改为“乐器”的修改操作赋值一个较低的成本值。基于这样的简单推理，系统会认为“Jim bought a musical instrument at the store.”（Jim在商店买了一个乐器）比“Jim bought a power tool at the store.”（Jim在商店买了一个电源工具）更符合上述查询。

## 基于后向链接推理的问答系统

上一节中介绍的基于超图匹配的问答虽然能回答不少的查询，但是随着知识库Atomspace的不断变大，搜索匹配的工作会变得越来越复杂，也越来越耗时。而且基于超图匹配的方法只是一个粗糙的启发式搜索过程，不适用于这些情况：对查询时间要求高、知识库数据不足或者在知识库中的答案需要经过一定推理才能找到等。例如，该方法无法将查询“Who bought a glockenspiel?”（谁买了钟琴？）从下面的句子中找到近似匹配，但是这些句子都在一定程度上暗示了此查询的答案。

* Bob brought his new instrument home from the mall and played it for his kids.（Bob从商场带了新乐器回家，并为他的孩子们演奏了几曲。）
* Ling-ling’s house is like a museum of obscure musical instruments.（玲玲的家像一个稀有乐器博物馆）
* Jack always buys whatever he sees in a magazine....On theﬂight back from Guilin,there was nothing for Jack to read but a catalog of weird musical instruments.（Jack总爱买任何他在杂志上看到的东西。在从桂林回来的飞机上，只给Jack提供了一本稀奇乐器相关的杂志）

从另一方面来说，经过在适当的知识库上推理，也不难从上面的句子中找出适合此查询的答案。例如，如果想从第二个句子“Ling-ling’s house is like a museum of obscure musical instruments.”中找出符合查询的答案，我们可通过如下推理得到：

Ling-ling's house is like a museum of obscure musical instruments

（玲玲的家像一个稀有乐器博物馆。）

The glockenspiel is obscure, and the glockenspiel is a musical instrument

（钟琴很稀有，而且钟琴是一种乐器。）

A museum of entities of type X, contains many instances of X

（有关 X 的博物馆，含有很多 X 的实例。）

If X is in Y's house, then often Y has bought X

（如果 Y 的家中有 X，那么通常 Y 买了 X。）

|-

It is non-trivially probable that Ling-ling has bought a glockenspiel

（玲玲很有可能买了一个钟琴。）

在Atomspace中，这样的定性推理可以通过不同很多方式进行。下面我们将解释其中一种方式的推理过程。推理的前提可表示如下：

Ling-ling's house is like a museum of obscure musical instruments：

（玲玲的家像一个稀有乐器博物馆。）

InheritanceLink ConceptNode "house@123" ConceptNode "house"

EvaluationLink PredicateNode "own" ConceptNode "Ling-ling" ConceptNode "house@123"

SimilarityLink ConceptNode "house@123" ConceptNode "museum@552"

InheritanceLink ConceptNode "museum@552"

ConceptNode "museum"

EvaluationLink PredicateNode "of" ConceptNode "museum@552"

ConceptNode "instruments@12d"

InheritanceLink

ConceptNode "instruments@12d" ConceptNode "obscure"

InheritanceLink

ConceptNode "instruments@12d" ConceptNode "musical"

InheritanceLink

ConceptNode "instrument@12d" ConceptNode "instrument"

The glockenspiel is obscure, and the glockenspiel is a musical instrument：

（钟琴很稀有，而且钟琴是一种乐器。）

InheritanceLink

ConceptNode "glockenspiel" ConceptNode "obscure"

InheritanceLink

ConceptNode "glockenspiel" ConceptNode "instrument@1c5"

InheritanceLink

ConceptNode "instrument@1c5" ConceptNode "musical"

InheritanceLink

ConceptNode "instrument@1c5" ConceptNode "obscure"

InheritanceLink

ConceptNode "instrument@1c5" ConceptNode "instrument"

A museum of entities of type X, contains many instances of X：

（有关 X 的博物馆，含有很多 X 的实例。）

ImplicationLink <.95,.99> ANDLink

InheritanceLink

$X

ConceptNode "museum" EvaluationLink PredicateNode "of"

$X

$Z EvaluationLink

PredicateNode "many"

SatisfyingSetLink ANDLink InheritanceLink

$A

$Z EvaluationLink

PredicateNode "in"

$A X

If X is in Y's house, then often Y has bought X：

（如果 Y 的家中有 X，那么通常 Y 买了 X。）

ImplicationLink <.8,.9> ANDLink

EvaluationLink

PredicateNode "in" ListLink

$X

$Z

InheritanceLink

$Z

ConceptNode "house" EvaluationLink PredicateNode "own" ListLink

$Y

$Z EvaluationLink

PredicateNode "buy"

ListLink

$Y

$X

根据上面的推理前提，PLN可以推理得到下面的结论：

EvaluationLink <s,c>

PredicateNode "buy"

ConceptNode "Ling-ling"

ConceptNode "glockenspiel"

真值 < s, c > 的大小不仅取决于推理过程中的使用到的推理规则的具体参数，还取决于节点本身的概率大小（“glockenspiel”节点的概率值用于表示钟琴在这世界上有多常见，这有助于估算钟琴是稀有乐器的概率）。因此，即使推理的前提给出很高的强度和置信度的真值如 < 1, .95 >，推断得出的结论的真值也可能很低如< .05, .6 >。不过由于推理的不确定性，得到的这样的低真值结论也是合理的。上面 例子给出的前提是简化过的，比如，没有给出这样的事实：“博物馆通常有稀有的东西”，可表示如下：

ImplicationLink <.3,.8>

ANDLink

InheritanceLink

$X

ConceptNode "museum"

EvaluationLink

PredicateNode "in"

$X

$Z

InheritanceLink

$Z

ConceptNode "obscure"

对该事实的考虑将在一定程度上增加结论的强度。总的来说，在推理链中不同阶段选择不同的前提，导致可以通过很多不同推理路径得到一个推理结论。每一条推理规则的执行是精确的，但对于选择哪条推理链则是一个非常模糊的问题，需要知识库Atomspace中的具体知识来指导。目前我们的逻辑系统PLN能完成上面的推理，但使用自然语言输入来进行这样的推理还存在一些问题。

## 一个综合的问答规划器

综合上述对问答规划的讨论，我们还针对信息查找问题设计并实现了一个合理的综合问答规划器，设计思路可参考图8.1。

其实现方法如下：

* 给定一个用于信息查找的问题Q，回答Q的动机，以及所允许的最长等待时间T；
* 如果T的值大，则试图通过逻辑推理来回答Q如果成功，则将得到的原子集合输入微观规划器用于应答如果失败，则通过基于超图的模糊匹配来回答Q；
* 如果成功，则将模糊匹配得到的原子集合来做为种子进行进一步推理，试图再次通过逻辑推理来回答Q；
* 如果成功，则将得到的原子集合输入微观规划器用于应答如果失败，则将模糊匹配得到的原子集合输入微观规划器用于应答；
* 如果失败，则触发StumpedByQuestion规划器；
* 如果T的值小，则试图通过浅层推理来快速回答Q；
* 如果成功，则将得到的原子集合输入微观规划器用于应答。

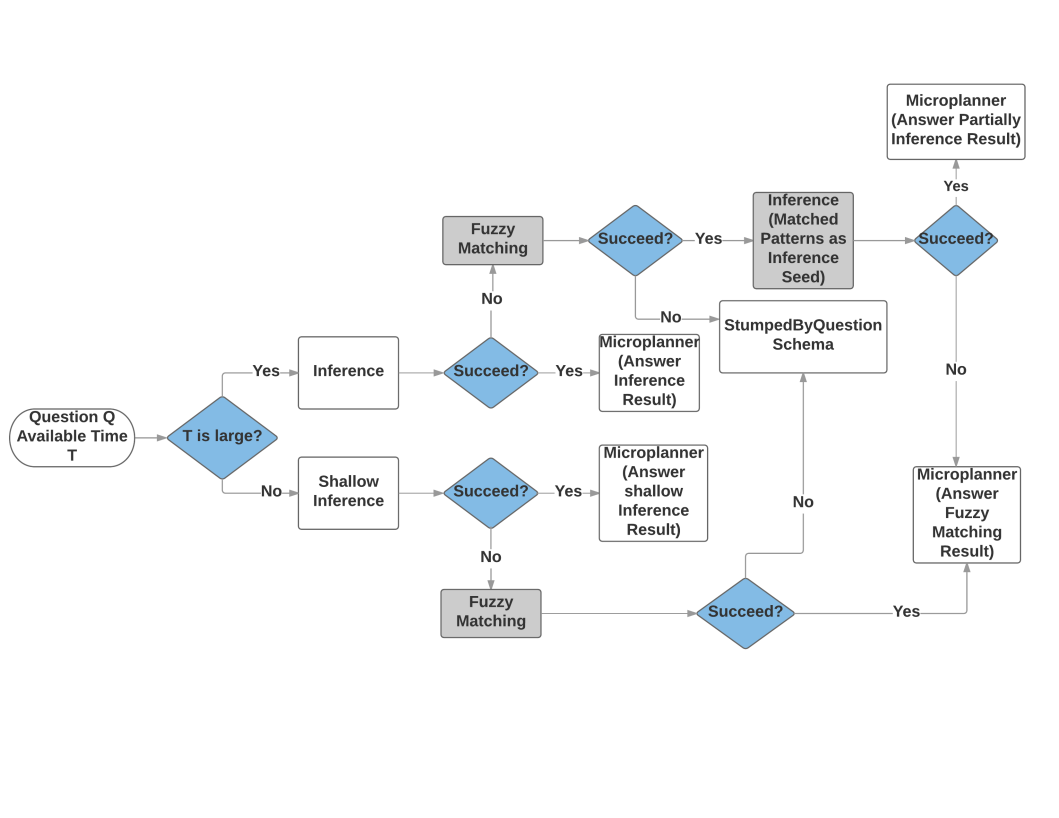


图 8.1 信息查找问答规划器的流程

如果失败，则试图通过基于超图的模糊匹配来回答 Q 如果成功，则将得到的原子集合输入微观规划器用于应答 如果失败，则触发 StumpedByQuestion 规划器

其中，StumpedByQuestion Schema 可规划如下：

StumpedByQuestion Schema：

给定一个问题 Q （其中 Q 为智能对话系统试图回答但未成功）

当 Q 的重要值低时，

触发 ExpressIgnorance 规划器

当 Q 的重要值很高时，

使用 PLN 中的后向链推理找到 P，使得 P ==> Q

其中，P 相对简洁，且上述蕴涵推理的真值很高

如果找到合适的 P，将 P 标记为一个问题，

并输入到微观规划器用于应答 如果没有找到合适的 P，则

触发 ExpressIgnorance 规划器

其中，ExpressIgnorance 规划器可规划如下：

ExpressIgnorance chema：

将用于表示"I don't know" 或者

"I don't know the answer to Q"的原子集合输入微观规划器用于应答

上面例子说明了 CogDial 方法的本质，我们可以将自适应的语篇管理能力分为下面三个阶段：

1. 用 Scheme 或者 Python 编写相应语篇管理行为代码，然后绑定在 Grounded- SchemaNodes 里。OpenPsi 可以在适当的时候调用这些语篇管理行为。
2. 用基于超图的知识表示形式 Atom 来表示语篇管理行为，通过“硬编码”设 置规划器参数，使其调用特定的认知处理程序（如微观规划器 Microplanner 或者后向链接推理工具 Backward Chainer 等）。
3. 根据经验，通过强化学习和模仿等方法来自动学习语篇管理行为。

很显然，第三种方法是我们最终想要的能进行智能灵活对话的认知对话系统，但也 无疑是最难的一种。我们 CogDial 首先实现第一种方法，然后在此基础上第二和第 三种方法。这种增量的开发方式使我们能逐步实现智能的对话处理。使用硬编码的 对话管理规划器，能使我们一种很直接的方式不断改进自然语言理解、生成和推理 系统中与对话相关的处理，而不是一开始就将这些可能不能处理复杂对话的自然语 言处理和推理系统集成到复杂的自动学习对话管理中。当这些模块逐渐完善后，我们将会使用自动学习得到的对话管理规则来替换我们目前使用的手工编写的对话管理规划器。

对于是非（真值查询）问题的应答，可用下面的类似方法来处理：

Composite Truth-value QA Schema：

给定一个真值查询问题 Q，回答该问题的动机，以及所允许的最长等待时间 T

试图通过逻辑推理来回答 Q 如果成功，则将得到的原子集合输入微观规划器用于应答 如果失败，试图通过基于超图的模糊匹配来回答 Q

如果成功，则将模糊匹配得到的原子集合来做为种子进行进一步推理，试图 再次通过逻辑推理来回答 Q

如果成功，则将得到的原子集合输入微观规划器用于应答 如果失败，则将模糊匹配得到的原子集合输入微观规划器用于应答

如果失败，则触发 StumpedByQuestion 规划器

Composite Choice QA Schema：

给定一个选择查询问题 Q，供选择的答案 C1-Ck，回答该问题的动机，以及所允许的最长等 待时机 T

重复以下步骤直到 T 被用完，或者找到一个置信度高的有效答案

对 k 个不同的查询“Ci 是否为 Q 的答案”进行并行评估，并对第 i 个查询赋上其相 关的资源 r\_i

使用 k 臂赌博机算法持续重新计算资源分配 r\_i

如果上述循环返回一个置信度高的有效答案，则将得到的原子集合输入微观规划器用于应答 如果上述循环完成，但没有返回置信度高的有效答案，则触发 StumpedByQuestion 规划器

## 实例分析

该认知对话系统目前还处于非常初步的阶段，并没有完全实现我们前面章节列 出的所有言语规划器和对话控制机制。目前还无法被用做一个很实用的对话系统， 但我们认为，在上述的理论分析研究基础上，在完整实现不同的对话控制机制和言 语规划器后，CogDial 的潜在推理和学习能力能使其成为一个有实用价值的智能会 话系统。在本文撰写时，该系统还主要集中在问答机制的实现。本小节给出了一些 目前已经实现的简单的对话结果。

**CogDial：** Hi there

所使用的言语规划器：开场白（Conventional-Opening）

Ruiting： Hi

**CogDial：** What do you want to talk about today?

所使用的言语规划器：开放问题（Open-Question）

**Ruiting：** What do YOU want to talk about today?

**CogDial：** I’m thinking.

所使用的言语规划器：自言自语（Self-talk）

**Ruiting：** Do people really seek happiness?

**CogDial：** Do you seek happiness?

* 所使用的言语规划器：是非问答（Yes-No Question）

**Ruiting：** Sometimes...

**CogDial：** Hmmm...

所使用的言语规划器：转移话题（Turn-Exit）

**Ruiting：** Do you have any questions?

**CogDial：** I’m tired of asking quesitons.

所使用的言语规划器：是非问答（Yes-No Question）

## 深层语义解析的挑战

在研究和设计智能对话系统中，最具挑战性的并不是上述的复杂推理过程。在 给定大知识库 Atomspace 的情况下，上述的推理过程会变得相当困难，但是我们可 以通过在 OpenCog 中实现的注意力经济分配算法（EConomic Attention AllocatioN, 以下简称 ECAN）[[96](#_bookmark215)] 来对推理过程进行实时动态规划和控制，虽然 ECAN 目前仍 需进一步的研究和改进，但从我们目前的实验数据来看，该方法是很可行的。在智 能对话系统中，比复杂的推理更具有挑战性的问题是，如何使 Atomspace 获取上述 问答规划器中的推理前提的超图表示形式。简单的说，这个挑战也就是如何将自然 语言转换成合理准确的语义解析，这恐怕也是实现高度智能化的自然语言对话系统 过程中最棘手的问题了。

例如，一个近乎准确的事实“如果 A 继承 Z，那么有很多 A 在 Z 的博物馆里” 可表示如下：

ImplicationLink <.95,.99>

ANDLink

InheritanceLink

$X

ConceptNode "museum" EvaluationLink

PredicateNode "of"

$X

$Z

EvaluationLink

PredicateNode "many" SatisfyingSetLink

ANDLink

InheritanceLink

$A

$Z EvaluationLink

PredicateNode "in"

$A

$X

当前的自然语言系统不太可能获取上述例子中想要表达的推理形式。目前的句法分析或语义分析工具也无法根据字典中对博物馆（museum）的解释（如下）来得到 上述的推理表达形式。

museum： a building in which objects of historical, scientific, artistic, or cul- tural interest are stored and exhibited.

（博物馆：珍藏和陈列历史、科学、艺术或者文化遗产的建筑）

但是上述的推理形式可以从自然语言处理系统可以理解的各种句子中推理得出，例 如：

The New York Museum of Modern Art has one of the world's largest collection- s of French Impressionist works

（纽约的现代艺术博物馆是全球收藏法国印象派作品最多的博物馆之一。）

The British Museum boasts the world's largest Egyptology exhibit, with over five hun- dred sarcophagi on display,

along with some of the most perfectly preserved examples of engraved hieroglyphics.

（大英博物馆以全球珍藏最多古埃及文物闻名。其中有超过 500 个石棺，有些还带有保存最 完整的象形雕刻文字。）

The US Postal Service Museum hosts enough rare stamps to satisfy even the most avid n- umismatist, but also

a variety of surprisingly interesting displays recounting, for example, the roles played by var-

ious animals in

the history of the postal service.

（美国邮政博物馆拥有足够多的稀有邮票，足以满足即便是最热心的徽章收藏家。不仅如 此，该博物馆还展示很多出奇又有趣的历史回放，比如：各种动物扮演的在邮政历史上的 不同角色。）

通过这些句子提供的信息，得到上述的推理形式就不是特别困难。只需要将这些 “大（large）”“足够（enough）”“各种（variety）”等这些量词转换成一般概念层 次上的“很多（many）”。而这样的关联完全可以从以下类似的句子中推理得出：

The New York Museum of Modern Art has one of the world's largest collection- s of French Impressionist works.

Many of these works are worth tens of millions of dollars, but their artistic val-

ue is incalculable.

（纽约的现代艺术博物馆是全球收藏法国印象派作品最多的博物馆之一。

其中很多作品都值上千万美金，但是他们的艺术价值是无价的。）

The dogs had more than enough steaks this morning. Many of them were rot- ten but they sure didn't seem to mind.

（狗狗们今天上午吃了太多牛排。其中很多牛排已经馊掉了，但是狗狗似乎一点也不介意。）

这些例句能很明确地将其他的量词和“很多（many）”联系起来。例如，解析上面 最后一个例句后，RelEx2Logic 输出的语义逻辑关系中含有：

EvaluationLink

PredicateNode "enough" ConceptNode "steaks@5v2"

EvaluationLink

PredicateNode "many" ConceptNode "steaks@5v2"

在解析上面列出的第一个句子后，RelEx2Logic 输出的语义逻辑关系中含有：

ImplicationLink

PredicateNode "largest@737" PredicateNode "largest"

EvaluationLink

PredicateNode "largest@737" ConceptNode "collection@a34"

EvaluationLink

PredicateNode "of" ConceptNode "collection@a34" ConceptNode "works@8fw"

EvaluationLink

PredicateNode "many" ConceptNode "works@8fw"

因此如果系统的知识库中含有以下推理常识：

ImplicationLink

AndLink

AndLink

InheritanceLink

$C

ConceptNode "collection" EvaluationLink

PredicateNode "of"

ConceptNode "collection"

$W EvaluationLink

PredicateNode "many"

$W

EvaluationLink

PredicateNode "many"

$C

那么就能推断出：

ImplicationLink

PredicateNode "largest@737" PredicateNode "largest"

EvaluationLink

PredicateNode "largest@737" ConceptNode "collection@a34"

EvaluationLink

PredicateNode "many" ConceptNode "collection@a34"

如果系统中还有下列常识：

ImplicationLink

PredicateNode "largest" PredicateNode "large"

那么就能推断出：

EvaluationLink

PredicateNode "large" ConceptNode "collection@a34"

EvaluationLink

PredicateNode "many" ConceptNode "collection@a34"

Putting these conclusions about enough vs. many and large vs. many, from these sentences, together with similar conclusions from other sentences, the (uncertain) conclu- sions

将这些从上面例句中推理得来的“大（large）”“足够（enough）”和“很多（many）”相关的结论联合起来，不难得出下列（非确定性）结论：

ImplicationLink

EvaluationLink

PredicateNode "large" ConceptNode $X

EvaluationLink

PredicateNode "many" ConceptNode $X

ImplicationLink

EvaluationLink

PredicateNode "enough" ConceptNode $X

EvaluationLink

PredicateNode "many"

ConceptNode $X

这些可以看成是非确定性的启发式的蕴涵规则，而这些不确定的直观的知识也正是很多常识性推理的基础。

上述推理中我们假设系统的知识库已存在一定的推理前提，如：

ImplicationLink

AndLink

InheritanceLink

$C

ConceptNode "collection"

EvaluationLink

PredicateNode "of"

ConceptNode "collection"

$W

EvaluationLink

PredicateNode

"many"

$W

AndLink

EvaluationLink

PredicateNode "many"

$C

那么系统如何获取这些假定的直观常识呢，我们的猜想是可以通过自然语言理解系统解析一些相关句子得到，对于本例，系统可以通过解析下面句子获取相关常识：

His stamp collection is bigger than hers.

（他的邮票收藏库比她的大。）

The bigger bowl has many kinds of candy in it.

（大一点的那个碗里有很多种糖果。）

其中第一个句子中的“collection（收藏）”与“bigger（更大）”关联，而第二个句子中“bigger（更大）”与“many（很多）”关联，因此通过演绎推理可以得出“collection（收藏）”与“many（很多）”关联。一般来说，解析某些特定句子所需要的知识都能从其他的句子中获得，但解析这些“其他的句子”可能需要从另外的其他的句子中获取相关知识。因此我们面临的挑战是如何构建一个强大的系统框架，使得获取推理常识的知识网络是一个良性而不是恶性循环。

这个挑战其中一部分是由组合爆炸引起，在我们的超图知识库Atomspace中包括很多不同句子的解析结果，这很容易导致过量的不同推断。上述例子中的任何特定推断理论上都可以通过目标驱动的后向链接推理得出，如8.2中讨论的回答查询问题的例子“谁在商店买钟琴”中的推理模式，这样一来，我们需要设置Atomspace能储存相对久的知识以保证推理的时候有足够多的推理前提。同样，上述例子中的任何特地推断原则上也可以通过系统自动搜索相关有趣信息来进行前向链接推理得出。不论用哪一种推理方法，不仅需要一个非常丰富复杂的知识库的支持，如果合理高效地进行推理控制更是巨大的研究挑战。

这个挑战并不仅仅是针对查询处理，其实，更关键挑战基本上可以归结为常识推理的问题上[100]。考虑到即使是一个简单的常识推理，所涉及的知识领域和其复杂性不言而喻，纯粹通过手工编码构建的复杂知识库如Cyc[6]已被证实并不可行。但是随着大数据处理技术的发展，通过分析和挖掘自然语言语料库以及智能体的涉身经验中的语言和非语言信息的综合数据库，来获取常识推理所需要的知识便成为一个更可行有效的方法。然而，根据自然语言的特性，某一个自然语言句子中用于推理的逻辑表现形式，常常与其他句子的表现形式不尽相同，因此通常需要一些从额外的句子中获取的常识来对这些句子进行逻辑形式重排，这也就要求一个可靠的能进行实时不确定逻辑推理系统。这无疑是一个相当棘手的问题。

要解决上述的常识知识问题，对于智能对话系统来说，其中一种方法是粗略模仿人类儿童的语言和认知成长轨迹。也就是说，从相对简单的知识库知识库着手，使得智能对话系统能通过知识库中相对简单的句子和人类交流相对简单的知识。当系统已经建立了基于这些相对简单的知识网络下的常识性知识后，再开始对系统输入稍微复杂的知识–等等，依此下去，这样的对话系统很可能会发展成一个拥有相当的常识语义库并能做出实时合理应答的智能对话系统。从概念上讲，上述方法和[121]中的无监督语言学习类似，即通过逐步将越来越复杂的自然语言句子输入系统，使得系统逐步理解越来越复杂的句子。[122]也提出了类似的涉身行为学习。但我们这里提出的观点更针对智能对话系统，更注重如何从自然语言中抽取出有用的合理的语义知识等的学习。

## 本章小结

目前我们的系统还不足以达到符合人类的智能标准，但这一个介于目前相关研究水平和人类水平之间的对话系统可以看做是使用更可行的研究方法在智能对话系统中集成并完善更多的功能，而不是一味盲目地去追求达到人类水平的目标。

本章充分利用了我们的研究成果，在第4章中提出的认知对话系统CogDial的概念模型基础上，并结合第5章中讨论的自然语言理解系统、第6章中讨论的自然语言生成系统，以及2.3中讨论的PLN推理系统，以认知模型OpenPsi中的动机机制等技术来设计并实现了几个合理有效的对话管理机制。本章还讨论了我们在智能对话系统的研究中发现的深层语义解析的挑战，也给未来的研究提供了一个明确的方向。下一步工作将进一步完善本章中提出的各种对话管理机制并开发更多新颖智能的对话控制体系结构，并最终运用到机器人或者其他智能体软件系统上。

# 

第九章 总结与展望

构建一个人类水准的对话系统是一项艰巨而长远的任务，其深度和内容都已经远超本文所能涵盖的范围。本文针对当前对话系统中的认知技术研究的不足，深入研究了认知对话系统中所需涉及的认知技术的范畴，在这些认知需求的基础上提出了一个动机驱动的认知对话控制模型，我们认为，一个自然的人机对话系统，必须遵循以下四个原则：知识深层表示、不确定性逻辑推理、语言与涉身知识的融合以及动机驱动的对话控制。本文着眼于未来对话系统中认知技术的发展，结合这四项原则提出了一个全新的动机驱动的认知对话系统模型，并以言语行为理论作为指导将语言和涉身知识融合起来使得对话过程更人性化；设计并实现了一个将自然语言的依存句法关系输出转换成基于超图的抽象逻辑表示的自然语言理解框架；在概率逻辑网络的逻辑架构上应用不确定性逻辑推理，实现了在自然语言语句上执行的常识推理；设计并实现了一个基于超图匹配的将抽象的语义超图表示转换成自然语言的表层生成框架；在上述几项研究基础上，设计并实现了一个以信息查询为动机的问答系统。 作为总结，我们将在本章指出本文工作的主要贡献与创新点，并提出未来认知对话系统研究的建议和展望。



## 本文的主要贡献和创新

本文的主要贡献与创新表现在以下几个方面：

（1）动机驱动的对话控制模型

（2）语言和涉身知识的融合

（3）自然语言理解

（4）自然语言生成

（5）基于自然语言的概率逻辑推理

（6）基于概率逻辑网的信息查询问答系统

## 下一步研究方向

（1）进一步完善言语行为规划器

（2）使用概率逻辑网对认知对话的全局控制进行规划和实验

（3） 提高自然语言理解和自然语言生成系统的准确率和效率

（4）无监督的语言学习

# 

# 参考文献

[1] A. Turing. Computing machinery and intelligence[J]. Mind. 1950. **59**

[2] J. Weizenbaum. ELIZA—a computer program for the study of natural language communication between man and machine[J]. Communications of the ACM. 1966. **9**(1)：36–45

[3] S. Arora, K. Batra, S. Singh. Dialogue System： A Brief Review[J]. CoRR. 2013. **abs/1306.4134**.

URL [http：//arxiv.org/abs/1306.4134](http://arxiv.org/abs/1306.4134)

[4] D. Jurafsky, J. H. Martin. Speech and language processing ： an introduction to natural language processing, computational linguistics, and speech recognition[M]. Upper Saddle River, N.J.： Pearson Prentice Hall, 2009. ISBN 9780131873216 0131873210. URL [http：//www.amazon.com/Speech-Language-Processing-2nd Edition/dp/0131873210/ref=pd\_bxgy\_b\_img\_y](http://www.amazon.com/Speech-Language-Processing-2nd-Edition/dp/0131873210/ref%3Dpd_bxgy_b_img_y)

[5] Y. Gal. Semantics, Modelling, and the Problem of Representation of Meaning – a Brief Survey of Recent Literature[R]. Tech. rep., University of Cambridge. 2013

[6] D. Lenat, R. V. Guha. Building Large Knowledge-Based Systems： Representation and Inference in the Cyc Project[M]. Addison-Wesley, 1990

[7] S. Piantadosi, N. Goodman, B. A. Ellis, J. Tenenbaum. A Bayesian model of the acquisition of compositional semantics[C]. Proceedings of the thirtieth annual conference of the cognitive science society. 2008, 1620–1625

[8] B. T. S. Atkins. The role of the example in a frame semantics dictionary[M]. Submitted

[9] P. Liang, M. I. Jordan, D. Klein. Learning dependency-based compositional semantics[J]. Com- putational Linguistics. 2013. **39**(2)：389–446

[10] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor. Freebase： A Collaboratively Created Graph Database for Structuring Human Knowledge[C]. Proceedings of the 2008 ACM SIG- MOD International Conference on Management of Data. SIGMOD ’08, New York, NY, USA： ACM, 2008. ISBN 978-1-60558-102-6, 1247–1250. doi：10.1145/1376616.1376746. URL [http：//doi.acm.org/10.1145/1376616.1376746](http://doi.acm.org/10.1145/1376616.1376746)

[11] P. Liang, M. I. Jordan, D. Klein. Learning Dependency-based Compositional Semantics[C].

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistic- s： Human Language Technologies - Volume 1. HLT ’11, Stroudsburg, PA, USA： Associa- tion for Computational Linguistics, 2011. ISBN 978-1-932432-87-9, 590–599. URL [http：//dl.acm.org/citation.cfm?id=2002472.2002547](http://dl.acm.org/citation.cfm?id=2002472.2002547)

[12] L. Libkin. Database Theory — ICDT 2001： 8th International Conference London, UK, Jan- uary 4–6, 2001 Proceedings[M], chap. Expressive Power of SQL. Berlin, Heidelberg： Springer Berlin Heidelberg, 2001. ISBN 978-3-540-44503-6, 1–21. doi：10.1007/3-540-44503-X\_1. URL [http：//dx.doi.org/10.1007/3-540-44503-X\_1](http://dx.doi.org/10.1007/3-540-44503-X_1)

[13] A. Giordani, A. Moschitti. Semantic mapping between natural language questions and SQL queries via syntactic pairing[C]. Natural language processing and information systems.

Springer, 2009, 207–221

[14] A. Giordani, A. Moschitti. Corpora for Automatically Learning to Map Natural Language Ques- tions into SQL Queries.[C]. LREC. 2010

[15] A. Giordani, A. Moschitti. Syntactic structural kernels for natural language interfaces to databas- es[C]. Machine Learning and Knowledge Discovery in Databases. Springer, 2009, 391–406

[16] R. Speer, C. Havasi. Representing General Relational Knowledge in ConceptNet 5[C]. N. C. C. Chair), K. Choukri, T. Declerck, M. U. Do an, B. Maegaard, J. Mariani, A. Moreno, J. Odijk,

S. Piperidis (Editors) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12). Istanbul, Turkey： European Language Resources Association (EL-RA), 2012. ISBN 978-2-9517408-7-7

[17] H. Liu, P. Singh. ConceptNet - a practical commonsense reasoning tool-kit[J]. BT technology journal. 2004. **22**(4)：211–226

[18] H. Poon, P. Domingos. Unsupervised Semantic Parsing[C]. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. Singapore： Association for Computa- tional Linguistics, 2009, 1–10. URL [http：//www.aclweb.org/anthology/D/D09/D09-1001](http://www.aclweb.org/anthology/D/D09/D09-1001)

[19] P. Domingos, M. Richardson. 1 Markov Logic： A Unifying Framework for Statistical Relational Learning[J]. Statistical Relational Learning. 2007：339

[20] I. Titov, A. Klementiev. A Bayesian model for unsupervised semantic parsing[C]. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics： Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011, 1445–1455

[21] K. Kersting, L. De Raedt. 1 Bayesian Logic Programming： Theory and Tool[J]. Statistical Relational Learning. 2007：291

[22] M. Brocheler, L. Mihalkova, L. Getoor. Probabilistic similarity logic[J]. arXiv preprint arXiv： 1203.3469. 2012

[23] L. De Raedt, K. Kersting. Probabilistic logic learning[J]. ACM SIGKDD Explorations Newslet- ter. 2003. **5**(1)：31–48

[24] P. Wang. Rigid Flexibility： The Logic of Intelligence[M]. Springer, 2006

[25] M. Baroni, R. Zamparelli. Nouns are vectors, adjectives are matrices： Representing adjective- noun constructions in semantic space[C]. Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2010, 1183–1193

[26] E. Grefenstette. Towards a formal distributional semantics： Simulating logical calculi with tensors[J]. arXiv preprint arXiv：1304.5823. 2013

[27] J. Peckham. A new generation of spoken dialogue systems： Results and lessons from the SUNDI- AL project[C]. Third European Conference on Speech Communication and Technology. 1993, 33−40

[28] L. Lamel, S. Rosset, J. L. Gauvain, S. Bennacef. The LIMSI ARISE system for train travel information[C]. Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on, vol. 1. 1999. ISSN 1520-6149, 501–504 vol. 1. doi：10.1109/ ICASSP.1999.758172

[29] M. M. University, M. F. Mctear. Modelling spoken dialogues with state transition diagrams： experiences with the CSLU toolkit[C]. Proc 5th International Conference on Spoken Language Processing. 1998, 1223–1226

[30] C. Rich, C. L. Sidner. COLLAGEN： A Collaboration Manager for Software Interface Agents[J].

User Modeling and User-Adapted Interaction. 1998. **8**(3)： 315–350. ISSN 1573-1391. doi： 10.1023/A：1008204020038. URL http：//dx.doi.org/10.1023/A：1008204020038

[31] D. Bohus, A. I. Rudnicky. Ravenclaw： dialog management using hierarchical task decom- position and an expectation agenda. [C]. INTERSPEECH. ISCA, 2003, 597−600. URL [http：//dblp.uni-trier.de/db/conf/interspeech/interspeech2003.html#BohusR03](http://dblp.uni-trier.de/db/conf/interspeech/interspeech2003.html#BohusR03)

[32] T. H. Bui, M. Rajman, M. Melichar. Text, Speech and Dialogue： 7th International Con- ference, TSD 2004, Brno, Czech Republic, September 8-11, 2004. Proceedings[M], chap. Rapid Dialogue Prototyping Methodology. Berlin, Heidelberg： Springer Berlin Heidelberg, 2004. ISBN 978-3-540-30120-2, 579–586. doi：10.1007/978-3-540-30120-2\_73. URL [http：//dx.doi.org/10.1007/978-3-540-30120-2\_73](http://dx.doi.org/10.1007/978-3-540-30120-2_73)

[33] S. Larsson, D. R. Traum. Information State and Dialogue Management in the TRINDI Dialogue Move Engine Toolkit[J]. Nat. Lang. Eng. Sep. 2000. **6**(3-4)：323–340. ISSN 1351-3249. doi： 10.1017/S1351324900002539. URL [http：//dx.doi.org/10.1017/S1351324900002539](http://dx.doi.org/10.1017/S1351324900002539)

[34] D. Bohus, A. I. Rudnicky. The RavenClaw dialog management framework： Architecture and systems[J]. Computer Speech & Language. 2009. **23**(3)：332–361

[35] S. Roy, L. V. Subramaniam. Automatic generation of domain models for call centers from noisy transcriptions[C]. Proceedings of the 21st International Conference on Computational Linguis- tics and the 44th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2006, 737–744

[36] S. Bangalore, G. Di Fabbrizio, A. Stent. Learning the structure of task-driven human-human dialogs[C]. in Proceedings of the Association for Computational Linguistics. Association for Computational Linguistics, Association for Computational Linguistics, 2006, 201–208

[37] C. Lee, S. Jung, K. Kim, G. G. Lee. Automatic Agenda Graph Construction from Human-Human Dialogs using Clustering Method[C]. Human Language Technologies： Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA, Short Papers. 2009, 89–92. URL [http：//www.aclweb.org/](http://www.aclweb.org/anthology/N09-2023) [anthology/N09-2023](http://www.aclweb.org/anthology/N09-2023)

[38] D. Griol, G. Riccardi, E. Sanchis. Learning the structure of human-computer and human-human dialogs.[C]. INTERSPEECH. 2009, 2775–2778

[39] J. D. Williams, S. Young. Partially observable Markov decision processes for spoken dialog systems[J]. Computer Speech & Language. 2007. **21**(2)：393–422

[40] S. Young, M. Gasic, B. Thomson, J. D. Williams. Pomdp-based statistical spoken dialog systems： A review[J]. Proceedings of the IEEE. 2013. **101**(5)：1160–1179

[41] J. Williams, A. Raux, D. Ramachandran, A. Black. The dialog state tracking challenge[C].

Proceedings of the SIGDIAL 2013 Conference. 2013, 404–413

[42] S. Kim, L. F. D’Haro, R. E. Banchs, J. D. Williams, M. Henderson. The fourth dialog state tracking challenge[C]. Proceedings of the 7th International Workshop on Spoken Dialogue Systems (IWSDS). 2016

[43] M. Henderson, B. Thomson, S. Young. Deep neural network approach for the dialog state track- ing challenge[C]. Proceedings of the SIGDIAL 2013 Conference. 2013, 467–471

[44] T. Paek. Reinforcement learning for spoken dialogue systems： Comparing strengths and weak- nesses for practical deployment[C]. Proc. Dialog-on-Dialog Workshop, Interspeech. 2006

[45] 俞凯, 陈露, 陈博, 孙锴, 朱苏. 任务型人机对话系统中的认知技术——概念, 进展及其未 来[J]. 计算机学报. 2015. **38**(12)：2333–2348

[46] J. D. Williams, S. Young. Scaling Up POMDPs for Dialog Management： The ”Summary POMDP” Method[C]. Proc IEEE Workshop on Automatic Speech Recognition and Under- standing (ASRU), San Juan, Puerto Rico, USA. 2005. URL [http：//research.microsoft.](http://research.microsoft.com/apps/pubs/default.aspx?id=160937) [com/apps/pubs/default.aspx?id=160937](http://research.microsoft.com/apps/pubs/default.aspx?id=160937)

[47] O. Lemon, K. Georgila, J. Henderson. Evaluating eﬀectiveness and portability of reinforce- ment learned dialogue strategies with real users： the TALK TownInfo evaluation[C]. Spoken Language Technology Workshop, 2006. IEEE. IEEE, 2006, 178–181

[48] S. Young, J. Schatzmann, K. Weilhammer, H. Ye. The hidden information state approach to dialog management[C]. Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, vol. 4. IEEE, 2007, IV–149

[49] B. Thomson, J. Schatzmann, S. Young. Bayesian update of dialogue state for robust dialogue systems[C]. Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on. IEEE, 2008, 4937–4940

[50] J. D. Williams. The best of both worlds： unifying conventional dialog systems and POMDPs. [C]. INTERSPEECH. 2008, 1173–1176

[51] L. F. Hurtado, D. Griol, E. Segarra, E. Sanchis. A Stochastic Approach for Dialog Manage- ment based on Neural Networks[C]. IEEE Workshop on Automatic Speech Recognition and Understanding, 2005. 2005, 226−231. doi：10.1109/ASRU.2005.1566518

[52] H. Murao, N. Kawaguchi, S. Matsubara, Y. Yamaguchi, Y. Inagaki. Example-based spoken dialogue system using WOZ system log[C]. In： Proc. 4th SIGDIAL Workshop on Discourse and Dialogue. 2003, 140–148

[53] N. Inui, T. Ebe, B. Indurkhya, Y. Kotani. A case-based natural language dialogue system using dialogue act[C]. Systems, Man, and Cybernetics, 2001 IEEE International Conference on, vol. 1.

IEEE, 2001, 193–198

[54] C. Lee, S. Jung, S. Kim, G. G. Lee. Example-based dialog modeling for practical multi-domain dialog system[J]. Speech Communication. 2009. **51**(5)：466–484

[55] L. Shang, Z. Lu, H. Li. Neural responding machine for short-text conversation[J]. arXiv preprint arXiv：1503.02364. 2015

[56] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, B. Dolan. A neural network approach to context-sensitive generation of conversational responses[J]. arXiv preprint arXiv：1506.06714. 2015

[57] O. Vinyals, Q. Le. A neural conversational model[J]. arXiv preprint arXiv：1506.05869. 2015 [58] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, J. Pineau. Building End-To-End Dia-

logue Systems Using Generative Hierarchical Neural Network Models[C]. Proceedings of the

Thirtieth AAAI Conference on Artiﬁcial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA. 2016, 3776–3784. URL [http：//www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/](http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11957) [view/11957](http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11957)

[59] O. Pietquin, T. Dutoit. A probabilistic framework for dialog simulation and optimal strategy learning[J]. Audio, Speech, and Language Processing, IEEE Transactions on. 2006. **14**(2)： 589–599

[60] J. Schatzmann, B. Thomson, S. Young. Error simulation for training statistical dialogue sys- tems[C]. Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on.

IEEE, 2007, 526–531

[61] J. Henderson, O. Lemon, K. Georgila. Hybrid reinforcement/ supervised learning of dialogue policies from ﬁxed data sets[J]. Computational Linguistics. 2008. **34**(4)：487–511

[62] C. Lee, S. Jung, K. Kim, G. G. Lee. Hybrid approach to robust dialog management using agenda and dialog examples[J]. Computer speech & language. 2010. **24**(4)：609–631

[63] P. Lison. A hybrid approach to dialogue management based on probabilistic rules[J]. Computer Speech & Language. 2015. **34**(1)：232–255

[64] C.-p. Wu, L. S. Hsu, C. L. Tan. A survey on statistical approaches to natural language process- ing[M]. Citeseer, 1992

[65] V. Gupta. A Survey of Natural Language Processing Techniques[J]. International Journal of Computer Science & Engineering Technology. 2014. **5**(1)：14–16. ISSN 2229-3345

[66] J. M. Eisner. Three new probabilistic models for dependency parsing： An exploration[C]. Pro- ceedings of the 16th conference on Computational linguistics-Volume 1. Association for Com- putational Linguistics, 1996, 340–345

[67] J. Eisner. An empirical comparison of probability models for dependency grammar[J]. arXiv preprint cmp-lg/9706004. 1997

[68] H. Yamada, Y. Matsumoto. Statistical dependency analysis with support vector machines[C].

Proceedings of IWPT, vol. 3. 2003, 195–206

[69] N. Chomsky. Syntactic Structures[M]. The Hague： Mouton, 1957

[70] C. Pollard, I. A. Sag. Head-driven phrase structure grammar[M]. University of Chicago Press, 1994

[71] H. Thompson. Chart parsing and rule schemata in PSG[C]. Proceedings of the 19th annual meet- ing on Association for Computational Linguistics. Association for Computational Linguistics, 1981, 167–172

[72] E. Charniak. Statistical parsing with a context-free grammar and word statistics[J]. AAAI/IAAI.1997. 2005(598-603)：18

[73] M. Takaki, Y. Minoru, T. Kentaro, T. Jun’ichi. LiLFeS： towards a practical HPSG parser[C].

Proceedings of the 17th international conference on Computational linguistics-Volume 2. Asso- ciation for Computational Linguistics, 1998, 807–811

[74] E. Charniak, M. Johnson, M. Elsner, J. Austerweil, D. Ellis, I. Haxton, C. Hill, R. Shrivaths,

J. Moore, M. Pozar, et al. Multilevel coarse-to-ﬁne PCFG parsing[C]. Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. Association for Computational Linguistics, 2006, 168–175

[75] L. Tesnière. Eléments de Syntaxe Structurale[M]. Paris： Klincksieck, 1959

[76] A. Cheyer, D. Guzzoni. Method and apparatus for building an intelligent automated assis- tant[Z]. May 3 2007. US Patent App. 11/518,292, URL [http：//www.google.com/patents/](http://www.google.com/patents/US20070100790) [US20070100790](http://www.google.com/patents/US20070100790)

[77] D. Mirkovic, L. Cavedon. Dialogue management using scripts[Z]. Oct. 18 2011. US Patent 8,041,570, URL [https：//www.google.com/patents/US8041570](https://www.google.com/patents/US8041570)

[78] W. C. Mann, S. A. Thompson. Rhetorical structure theory： Description and construction of text structures[M]. Springer, 1987

[79] M. Stone, C. Doran. Sentence planning as description using tree adjoining grammar[C]. Pro- ceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics.

Association for Computational Linguistics, 1997, 198–205

[80] M. Elhadad, J. Robin. Controlling Content Realization with Functional Uniﬁcation Gram- mars[C]. ASPECTS OF AUTOMATED NATURAL LANGUAGE GENERATION. Springer Verlag, 1992, 89–104

[81] C. M. Mathiessen, J. Bateman. Text Generation and Systemic-Functional Linguistics[J]. Lon- don： Pinter. 1991

[82] I. Langkilde, K. Knight. Generation that Exploits Corpus-Based Statistical Knowledge[C]. 36th Annual Meeting of the Association for Computational Linguistics and 17th International Con- ference on Computational Linguistics, COLING-ACL ’98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference. 1998, 704–710. URL [http：//aclweb.org/anthology/P/P98/P98-1116.pdf](http://aclweb.org/anthology/P/P98/P98-1116.pdf)

[83] T.-H. Wen, M. Gasic, D. Kim, N. Mrksic, P.-H. Su, D. Vandyke, S. Young. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking[J]. arXiv preprint arXiv：1508.01755. 2015

[84] A. H. Oh, A. I. Rudnicky. Stochastic language generation for spoken dialogue systems[C].

Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3. As- sociation for Computational Linguistics, 2000, 27–32

[85] F. Mairesse, M. Gaši , F. Jur í ek, S. Keizer, B. Thomson, K. Yu, S. Young. Phrase-based sta- tistical language generation using graphical models and active learning[C]. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Com- putational Linguistics, 2010, 1552–1561

[86] F. Mairesse, S. Young. Stochastic language generation in dialogue using factored language models[J]. Computational Linguistics. 2014：763–799

[87] J. L. Austin. How to do things with words[M], vol. 367. Oxford university press, 1975

[88] J. R. Searle. Speech acts： An essay in the philosophy of language[M]. Cambridge university press, 1969

[89] D. P. Twitchell, M. Adkins, J. F. Nunamaker, J. K. Burgoon. Using speech act theory to model conversations for automated classiﬁcation and retrieval[C]. Proceedings of the 9th International Working Conference on the Language-Action Perspective on Communication Modelling. 2004, 121–129

[90] E. Tulving, R. Craik. The Oxford Handbook of Memory[M]. Oxford U. Press, 2005

[91] B. Goertzel. Toward a Formal Deﬁnition of Real-World General Intelligence[C]. Proceedings of AGI-10. 2010

[92] B. Goertzel, I. G. M. Ikle, A. Heljakka. Probabilistic Logic Networks[M]. Springer, 2008

[93] G. Fauconnier, M. Turner. The Way We Think： Conceptual Blending and the Mind’s Hidden Complexities[M]. Basic, 2002

[94] M. Looks. Competent Program Evolution[M]. PhD Thesis, Computer Science Department,

Washington University, 2006

[95] B. Goertzel, C. P. Et Al. An Integrative Methodology for Teaching Embodied Non-Linguistic Agents, Applied to Virtual Animals in Second Life[C]. Proc.of the First Conf. on AGI. IOS Press, 2008

[96] B. Goertzel, J. Pitt, M. Ikle, C. Pennachin, R. Liu. Glocal memory： a design principle for artiﬁcial brains and minds[J]. Neurocomputing. Apr. 2010

[97] J. Bach. Principles of Synthetic Intelligence[M]. Oxford University Press, 2009

[98] B. Goertzel, M. Ikle, I. Goertzel, A. Heljakka. Probabilistic Logic Networks[M]. Springer, 2008 [99] W. Lawvere, S. Schanuel. Conceptual mathematics： a ﬁrst introduction to categories[M]. Cam-

bridge University Press, 1997

[100] B. Goertzel, L. Coelho, N. Geisweiller, P. Janicic, C. Pennachin. Real World Reasoning[M].

Atlantis Press, 2011

[101] B. Goertzel, R. Lian. A Probabilistic Characterization of Fuzzy Semantics[J]. Proc. of ICAI-10,

Beijing. 2010

[102] D. Dörner. Die Mechanik des Seelenwagens. Eine neuronale Theorie der Handlungsregula- tion[M]. Verlag Hans Huber, 2002. ISBN 345683814X

[103] B. Goertzel, C. Pennachin, N. Geisweiller. Engineering General Intelligence, Part 1： A Path to Advanced AGI via Embodied Learning and Cognitive Synergy[M]. Springer： Atlantis Thinking Machines, 2013

[104] J. Bach. A Framework for Emergent Emotions, Based on Motivation and Cognitive Modula- tors[J]. International Journal of Synthetic Emotions (IJSE). 2012. **3**(1)：43–63

[105] S. P. Franklin. Artiﬁcial minds.[J]. 1995

[106] B. Goertzel, C. Pennachin, N. Geisweiller. Engineering General Intelligence, Part 2： The Cog- Prime Architecture for Integrative, Embodied AGI[M]. Springer： Atlantis Thinking Machines, 2013

[107] Z. Cai, B. Goertzel, C. Zhou, Y. Zhang, M. JIang, G. Yu. Dynamics of a computational aﬀective model inspired by D rnerÕs PSI theory.[J]. Cognitive Systems Research. 2011

[108] B. Goertzel. The Hidden Pattern[M]. Brown Walker, 2006

[109] D. Sleator, D. Temperley. Parsing English with a Link Grammar[J]. Third International Work- shop on Parsing Technologies. 1993

[110] Y. Chi, Y. Xia, Y. Yang, R. R. Muntz. Mining closed and maximal frequent subtrees from databases of labeled rooted trees..[J]. IEEE Trans. Knowledge and Data Engineering. 2005

[111] R. Hudson. Language Networks. The new Word Grammar[M]. Oxford University Press, 2007 [112] J. Hobbs. Resolving pronoun references.[J]. Lingua. 1978：311Ð338

[113] V. Voloshin. Introduction to Graph and Hypergraph Theory[M]. Nova Science, 2009

[114] R. Lian, B. Goertzel, S. Ke, J. OÕNeill, K. Sadeghi, S. Shiu, D. Wang, O. Watkins, G. Yu.

Syntax-Semantic Mapping for General Intelligence： Language Comprehension as Hypergraph Homomorphism, Language Generation as Constraint Satisfaction[C]. Artiﬁcial General Intelli- gence： Lecture Notes in Computer Science Volume 7716. Springer, 2012

[115] W. Lechner. Ellipsis in Comparatives[M]. Studies in generative grammar, Moulton de Gruyter, 2004. ISBN 9783110181180. URL [http：//books.google.com.hk/books?id=JsqUHHYSXCIC](http://books.google.com.hk/books?id=JsqUHHYSXCIC)

[116] R. Bhatt, S. Takahashi. Winfried Lechner, Ellipsis in comparatives[J]. The Journal of Comparative Germanic Linguistics. 2011. **14**(2)： 139–171. ISSN 1383-4924. doi：10.1007/ s10828-011-9042-3. URL [http：//dx.doi.org/10.1007/s10828-011-9042-3](http://dx.doi.org/10.1007/s10828-011-9042-3)

[117] M. Grant. The Parsing and Interpretation of Comparatives： More than Meets the Eye[M]. 2013.

URL [http：//scholarworks.umass.edu/open\_access\_dissertations/689/](http://scholarworks.umass.edu/open_access_dissertations/689/)

[118] R. Izvorski. A DP -shell for comparatives[J]. Proceeding of CONSOLE III. 1995：99–121 [119] R. Zass, A. Shashua. Probabilistic graph and hypergraph matching[C]. Computer Vision and

Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008, 1–8

[120] J. O’Neill, B. Goertzel, S. Ke, R. Lian, K. Sadeghi, S. Shiu, D. Wang, G. Yu. Pattern mining for general intelligence： the FISHGRAM algorithm for frequent and interesting subhypergraph mining[C]. Artiﬁcial General Intelligence. Springer, 2012, 189–198

[121] V. I. Spitkovsky, H. Alshawi, D. Jurafsky. Breaking Out of Local Optima with Count Transforms and Model Recombination： A Study in Grammar Induction.[C]. EMNLP. 2013, 1983–1995

[122] B. Goertzel. A Pragmatic Path Toward Endowing Virtually-Embodied AIs with Human-Level Linguistic Capability[C]. IEEE World Congress on Computational Intelligence (WCCI), 2008

[123] Aymeric Puech and Stephen Muggleton. A comparison of stochastic logic programs and Bayesian logic programs. In Proceedings of the IJCAI-2003 Workshop on Learning Statistical Models from Relational Data, pages 121–129, 2003.

[124] Tuyen N Huynh and Raymond J Mooney. Discriminative structure and parameter learning for Markov logic networks. In Proceedings of the 25th international conference on Machine learning, pages 416–423. ACM, 2008.

[125] Lilyana Mihalkova, Tuyen Huynh, and Raymond J Mooney. Mapping and revising Markov logic networks for transfer learning. In AAAI, volume 7, pages 608–614, 2007.

[126] Lilyana Mihalkova and Raymond J Mooney. Bottom-up learning of Markov logic network structure. In Proceedings of the 24th international conference on Machine learning, pages 625–632. ACM, 2007.

[127] Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond Mooney. Montague meets Markov: Deep semantics with probabilistic logical form, 2013.

[128] Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. arXiv preprint arXiv:1003.4394, 2010.

[129] Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. A tensor-based factorization model of semantic compositionality. In Proceedings of NAACLHLT, pages 1142–1151, 2013.

[130] Karl Moritz Hermann, Edward Grefenstette, and Phil Blunsom. “Not not bad” is not “bad”: A distributional account of negation. arXiv preprint arXiv:1306.2158, 2013.

# 

# 博士期间发表的论文

[1] 第一作者, 第二作者, 第三作者. 文章名一. 刊物, 2009, 08：10-13.

[2] 第一作者, 第二作者, 第三作者. 文章名二. 刊物, 2009, 09：20-23.

# 

# 致 谢

谢谢父母

谢谢导师 谢谢朋友 谢谢同学 谢谢其他人

1. [http：//wiki.opencog.org](http://wiki.opencog.org/) [↑](#footnote-ref-2)
2. p′th power average 可定义为p√ Xp [↑](#footnote-ref-3)
3. http://www.hansonrobotics.com/robot/sophia/ [↑](#footnote-ref-4)
4. 此为本文的延伸论点，而非 SWBD-DAMSL 言语行为的一部分。 [↑](#footnote-ref-5)
5. 这部分并不包含在 SWBD-DAMSL 中，但这显然是有别于其他的问句形式，值得提出论述 [↑](#footnote-ref-6)
6. 图片来自 G. Schneider, “Learning to Disambiguate Syntactic Relations”Linguistik online 17, 5/03 [↑](#footnote-ref-7)
7. RelEx中使用的依存关系类型集合可参见[http：//wiki.opencog.org/w/Binary\_relations](http://wiki.opencog.org/w/Binary_relations) [↑](#footnote-ref-8)
8. RelEx中使用的词态属性集合可参见[http：//wiki.opencog.org/w/Word\_properties](http://wiki.opencog.org/w/Word_properties) [↑](#footnote-ref-9)
9. [https：//www.youtube.com/ watch?v=ii-qdubNsx0)](https://www.youtube.com/watch?v=ii-qdubNsx0)) [↑](#footnote-ref-10)
10. [http：//wiki.opencog.org/wikihome/index.php/Unsupervised\_Language\_Learning\_](http://wiki.opencog.org/wikihome/index.php/Unsupervised_Language_Learning_Experimentation) [Experimentation](http://wiki.opencog.org/wikihome/index.php/Unsupervised_Language_Learning_Experimentation) [↑](#footnote-ref-11)