

密级: _____



中国科学院大学
University of Chinese Academy of Sciences

博士学位论文

基于词典学习和结构映射的语义解析技术研究

作者姓名: _____ 陈波

指导教师: _____ 孙乐 研究员

中国科学院软件研究所

学位类别: _____ 工学博士

学科专业: _____ 计算机软件与理论

培养单位: _____ 中国科学院软件研究所

2018 年 06 月

Semantic Parsing based on Lexicon Learning and
Structure Mapping

by
Bo Chen

A thesis submitted to
The University of Chinese Academy of Sciences
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Institute of Software, Chinese Academy of Sciences

June, 2018

学位论文独创性声明

本人郑重声明：我所呈交的学位论文是本人在导师指导下进行的研究工作及所取得的研究成果。尽我所知，除了文中已经标注引用的内容外，本论文中不含其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中作了明确的说明或致谢。本人知道本声明的法律结果由自己承担。

作者签名：_____ 日期：_____

关于学位论文使用授权的说明

本人完全了解中国科学院软件研究所有关保留、使用学位论文的规定，即：中国科学院软件研究所有权保留送交论文的复印件，允许论文被查阅和借阅；可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

(保密的论文在解密后应遵守此规定)

作者签名：_____ 导师签名：_____ 日期：_____

基于词典学习和结构映射的语义解析技术研究

摘要

随着科技和社会的发展，人们对信息精准化和应用智能化的需求越来越迫切。这需要搜索引擎能够理解用户所提交的问题，并快速给出精准的答案。这也需要更多的智能应用能够理解用户口语所表述的指令，并根据指令给出相应的操作和服务。语义解析是实现上述目标的核心技术之一。

语义解析是将自然语言句子解析成计算机可计算的语义表示的过程。现阶段的语义解析技术大多基于组合语义的原则：一个句子的意思是句子中各个成分的意思的组合。因此语义解析一般包含两个核心模块：结构预测和语义落地。其中结构预测是将句子映射到结构化的表示，而语义落地是将句子中的各成分都映射到物理世界中存在的物体或者关系。简单来说，结构预测指的是句子中的主谓宾等成分分别是什么；而语义落地指的是主语和宾语对应到物理世界中的哪个物体，谓语对应到物理世界中的哪个关系。现阶段的语义解析主要存在如下挑战和问题：a) 语义落地部分词典覆盖度不足的问题；b) 结构预测部分结构不匹配的问题；c) 语义落地部分过于依赖词典的问题；d) 结构预测部分过分依赖文法或者模板的问题。

本论文针对语义解析中存在的挑战以及现有工作中存在的问题，从结构预测和语义落地两个方面开展了研究，研究成果主要包括：

1. 针对语义落地部分中词典覆盖度不足的问题，提出了两种相互独立的基于词典扩充学习的语义解析方法。一种是基于桥连接的词典学习语义解析方法，该方法在现有工作的基础上，在训练过程中自动构建新的词汇，并加以学习，构成新的词典。为了进一步提升词典的准确度，还设计了通用的词语-二元谓词特征模板，以及基于投票机制的核心词典获取方法。另一种是基于半监督词典学习的语义解析方法。该方法利用少量种子词汇和大量自由文本以及词典资源（如同义词词典），通过标签传播算法，从少量标注样本出发，学习大量未标注的词汇。实验结果表明，这两种方法都能有效提高词汇的覆盖度，学习到高质量的新词汇，进而有效提升基准语义解析系统的性能。
2. 针对结构预测部分中结构不匹配的问题，提出了一种基于句子重写的语义解析方法。该方法利用句子重写，对给定的句子，生成与原句子意义相同，且其结构与目标语义表示结构一致的新的句子。通过对结构不匹配问题的分析和归纳，总结了两种常见的结构不匹配问题，分别是 1-N 型不匹配和 N-1 型不匹配。针对 1-N 型不匹配问题，本文提出一种基于词典替换的句子重写方

法；针对 N-1 型不匹配问题，本文提出一种基于模板替换的句子重写方法。实验结果表明，该方法能有效解决结构不匹配问题，从而有效提升语义解析系统的性能。

3. 针对传统语义解析方法过于依赖高质量词汇、特定文法、模板、启发式规则和人工特征的问题，提出了一种端到端的语义图生成的语义解析方法。该方法综合利用语义图的语义表示优势和循环神经网络模型的强序列预测能力，将语义解析转化为词语序列到动作序列的翻译问题，其中动作序列是对语义图构建过程的编码。在该方法中，通过将句法和语义约束条件加入解码过程，进一步改善了性能。实验结果表明，基于该方法的语义解析器的性能有明显提升。

关键词：语义解析，词典学习，结构映射，句子重写，语义图生成

Semantic Parsing based on Lexicon Learning and Structure Mapping

Abstract

With the rapid development of technology and society, people are increasingly demanding information precision and application intelligence. This requires the search engine can understand the meaning of the user's submitted questions and quickly respond with precise answers; It also requires more and more intelligent applications can understand the instructions given by the user, and respond with operations and services. Semantic parsing is one of the core technology to achieve the above goals.

Semantic Parsing is the task of mapping natural language sentences into meaning representations, which can be executed by computers. Most semantic parsing methods to current are based on the principle of compositional semantics: the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them. This makes a semantic parser need two function modules: structure mapping and semantic grounding. Structure mapping module maps a sentence into a structural representation, and semantic grounding module links the components of a sentence to items in a real world. Specifically, structure mapping module needs to get the structure of a sentence, and figure out which part is the subject, which part is the object, and which part is the predicate. Semantic grounding needs to map the subject and the object into the corresponding entities in a real world. Semantic parsing mainly faces the following challenges and problems: a) low-coverage of the lexicon in semantic grounding; b) structure mismatch problem in structure mapping; c) the problem of heavily relying on the lexicon in semantic grounding; d) the problem of heavily relying on grammars or templates in structure mapping.

In this thesis, we focus on the problem of semantic parsing. We start from structure mapping and semantic grounding, and attempt to propose and implement approaches to some problems in semantic parsing. The main contributions are summarized as follows:

1. To solve the problem of low-coverage of lexicon in semantic grounding, we propose two separate semantic parsing methods based on lexicon extending

learning. One is a bridge based lexicon learning method for semantic parsing. This method is on the basis of existing work, and it can bring in new vocabularies during training and learn a new lexicon with high-coverage. Furthermore, we design a new word-predicate feature template and utilize voting to gain core vocabularies for more accurate lexicon. The other is a semi-supervised lexicon learning method for semantic parsing. This method can increase the lexicon coverage by exploiting easily obtained text corpora and lexical resources. The intuition behind this approach is that similar phrases should map to similar predicates. Thus the phrase similarity can be used to propagate known predicate mappings to unknown mappings. We evaluate these two approaches on two benchmark datasets. The results show that our systems can learn new vocabularies for improving the coverage of lexicon. Meanwhile, our methods can improve semantic parsing performance.

2. To solve the problem of structure mismatch problem in structure mapping, we propose a semantic parsing method based on sentence rewriting. This method can effectively resolve the mismatch problem by rewriting a sentence into a new form which has the same structure with its target logical form. Specifically, we propose two sentence-rewriting methods for two common types of mismatch: a dictionary-based method for 1-N mismatch and a template-based method for N-1 mismatch. We evaluate our sentence rewriting based semantic parser on a benchmark semantic parsing dataset. Experimental results show that our system outperforms the base system significantly, and generates logical forms more accurately and parses sentences more robustly.
3. To solve the problem of heavily relying on the lexicon in semantic grounding, and grammars or templates in structure mapping, we propose a novel semantic parsing approach: Sequence-to-Action. This approach can simultaneously leverage the advantages of semantic graph representation and the strong prediction ability of Seq2Seq models. Specifically, we model semantic parsing as an end-to-end semantic graph generation process. To achieve the above goal, we first design an action set which can encode the generation process of semantic graph. And then we design a RNN model which can generate the action sequence for constructing the semantic graph of a sentence. Finally we further enhance parsing by incorporating both structure and semantic constraints during decoding. Experiments validate the effectiveness of our method.

Keywords: Semantic Parsing, Lexicon Learning, Structure Mapping, Sentence Rewriting, Semantic Graph Generation

目 录

摘要	vii
Abstract	ix
目录	xiii
图形列表	xvii
表格列表	xix
第一章 绪论	1
1.1 研究背景与意义	1
1.2 语义解析任务介绍	4
1.3 本文的主要研究内容及组织结构	6
第二章 语义解析方法综述	9
2.1 概述	9
2.2 组合语义学和模型理论语义学	9
2.2.1 语义落地	10
2.2.2 结构预测	13
2.3 语义解析	14
2.3.1 语义表示	15
2.3.2 知识库数据组织形式	20
2.3.3 解析算法	20
2.3.4 学习算法	21
2.3.5 公开数据集	21
2.4 语义解析方法分类	26
2.4.1 基于规则的语义解析	27
2.4.2 基于组合文法的语义解析	28
2.4.3 基于翻译的语义解析	30
2.4.4 基于语义图的语义解析	30
2.4.5 基于序列预测的语义解析	31
2.5 近期语义解析的应用举例	32

2.5.1 语言指令控制的机器人	32
2.5.2 解代数应用题的机器人	32
2.5.3 计算机操作语言助手	33
2.6 本章小结	33
第三章 基于词典扩展学习的语义解析	35
3.1 本章引论	35
3.2 基于桥连接的词典学习的语义解析	38
3.2.1 基准语义解析器	38
3.2.2 基于桥连接的词典学习的语义解析	40
3.3 基于半监督的词典学习的语义解析	42
3.3.1 基于半监督的词典学习	42
3.3.2 基于扩充词典的语义解析	47
3.4 实验	47
3.4.1 基于桥连接的词典学习的语义解析的实验	48
3.4.2 基于半监督的词典学习的语义解析的实验	50
3.5 本章小结	54
第四章 基于句子重写的语义解析	57
4.1 本章引论	57
4.2 句子重写	61
4.2.1 基于字典替换的句子重写	61
4.2.2 基于模板替换的句子重写	62
4.3 基于句子重写的语义解析	64
4.3.1 基准语义解析器	64
4.3.2 打分函数	65
4.3.3 参数学习算法	66
4.3.4 特征	66
4.4 实验	68
4.4.1 实验设置	68
4.4.2 实验结果与分析	68
4.4.3 错误分析	71
4.5 本章小结	72

第五章 基于端到端语义图生成的语义解析	75
5.1 本章引论	75
5.2 端到端的语义图生成方法	77
5.2.1 用于语义图构建的动作集	77
5.2.2 词语序列-动作序列的神经网络模型	79
5.3 带约束条件的神经语义解析模型	81
5.3.1 训练	81
5.3.2 推导	82
5.3.3 在解码中加入约束条件	82
5.4 实验	83
5.4.1 实验设置	84
5.4.2 实验结果与分析	84
5.4.3 详细分析	86
5.4.4 错误分析	87
5.5 本章小结	88
第六章 总结与展望	91
6.1 工作总结	91
6.2 工作展望	92
参考文献	95
简历	115
攻读博士学位期间发表的学术论文	117
致 谢	119

图形列表

1.1	Watson 在关于问答比赛的节目《危险边缘》中战胜了人类选手	1
1.2	在某搜索引擎上查询“姚明的身高是多少”的查询结果	2
1.3	“姚明”的百科页面中的结构化数据栏	2
1.4	使用 Siri 查询球队的下一场比赛情况	3
1.5	自然语言处理所研究问题的不同层次	4
1.6	语义解析的一个例子	5
1.7	Freebase 中关于实体“Barack Obama”的相关知识	6
1.8	本文的组织结构	7
2.1	用世界中的状态来解释句子的例子	9
2.2	CCG 中的词汇的例子	11
2.3	GENLEX 中生成候选词汇的图例说明	11
2.4	利用 Google 搜索“Barack Obama”和“Honolulu”得到的结果	12
2.5	CCG 中的函数应用 (functional application) 规则，包括前向和后向两种	13
2.6	CCG 中的函数组合 (functional application) 规则，包括前向和后向两种	13
2.7	利用规则进行语义组合的例子	14
2.8	句子结构不同，但是基于 FrameNet 的语义表示相同的例子	16
2.9	句子及其对应的基于 lambda-表达式的语义表示的例子	18
2.10	lambda-表达式和 lambda-DCS 对于一个句子的表示的对比	18
2.11	句子“The boy wants to go”的几种等价的语义表示	19
2.12	几个不同句子拥有相同的 AMR 表示的示例	19
2.13	ATIS 数据集中有上下文的并用 lambda-表达式进行语义表示的例子	26
2.14	查询模板及利用模板进行语义表示的例子	31
2.15	使用 lambda-表达式标注指令的语义的例子	32
2.16	利用语义解析解代数应用题的例子	32
2.17	自然语言指令和对应的 bash 指令	33
3.1	基于组合语义的语义解析的例子	35

3.2 对比我们的方法和之前的方法	37
3.3 基于桥连接的词典学习的语义解析方法框架图	38
3.4 基于组合语义的语义解析方法的基本框架图	39
3.5 桥连接方法示例	40
3.6 基于半监督的词典学习的语义解析框架图	42
3.7 对比有利用打分的种子词汇和没有打分的种子词汇分别进行图传播的例子	44
3.8 用于图传播的图	45
4.1 语义解析的一个例子	57
4.2 对比之前的语义解析方法和我们的使用句子重写的语义解析方法 ..	58
4.3 针对结构不匹配问题，我们的方法与其他方法的对比	60
4.4 我们提出的基于句子重写的语义解析方法的框架图	65
5.1 我们方法的概览，其右侧是一个示例	76
5.2 例子展示：一个句子，以及该句子所对应的语义图和用来构建该语义图的动作序列	78
5.3 基于注意力机制的用于生成动作序列的循环神经网络模型，附带一个添加约束条件的控制器	79
5.4 逻辑表达式与动作序列之间相互转换的流程简介图	81
5.5 使用约束条件来过滤错误动作的示例	83

表格列表

2.1 Wikipedia 中的一个表格的例子	20
2.2 OVERNIGHT 数据集中各个领域的数据统计	25
3.1 PPDB-2.0 中与词语 “ <i>currency</i> ” 最相似的几个词语.....	46
3.2 在 WEBQUESTIONS 数据集上的 A 组实验.....	48
3.3 在 FREE917 数据集上的 A 组实验	48
3.4 在 WEBQUESTIONS 数据集上的 B 组实验.....	49
3.5 在 FREE917 数据集上的 B 组实验	49
3.6 在 WEBQUESTIONS 数据集上的 C 组实验.....	50
3.7 在 FREE917 数据集上的 C 组实验	50
3.8 若干复述模板对的例子	50
3.9 在 WEBQUESTIONS 数据集上的结果对比	51
3.10 在 FREE917 数据集上的结果对比	51
3.11 在 WEBQUESTIONS 上使用不同词典的结果对比	53
3.12 在 FREE917 上使用不同词典的结果对比	53
3.13 在 WEBQUESTIONS 数据集上，使用不同资源进行相似性度量后再进行词典学习的效果	54
3.14 在 WEBQUESTIONS 数据集上，我们方法所学到的几个新的词汇，以及该词汇的词汇质量打分	54
4.1 若干样例，每个样例包括一个词语，它的目标语义表示以及该词语在 Wiktionary 中的解释	61
4.2 基于字典替换的句子重写方法的样例	62
4.3 若干 N-1 型不匹配的例子	62
4.4 若干复述模板对的例子	63
4.5 来自 WikiAnswers 复述语料库中的两个问句-簇	64
4.6 我们的参数更新算法	67
4.7 我们的方法与之前的方法在 WEBQUESTIONS 数据集上的实验结果 ..	69
4.8 在 50 个存在结构不匹配问题的测试集上的测试结果对比	70
4.9 对比两种句子重写方法的效果	70

4.10 若干基于句子重写的语义解析器比基准语义解析器表现更好的例子 ······	71
4.11 系统出现错误的主要原因以及出现的比例和样例 ······	72
5.1 GEO 数据集和 ATIS 数据集上的测试结果 ······	85
5.2 OVERNIGHT 数据集上的测试结果 ······	86
5.3 在三个数据集上对比两种实体处理机制的结果 ······	87
5.4 在三个数据集上线性化逻辑表达式和动作序列的平均长度 ······	87
5.5 用于错误分析的几个样例 ······	88

第一章 絮论

1.1 研究背景与意义

随着社会的发展和科技的进步，人们的生活水平逐渐提高，对智能机器人和智能应用的需求越来越大。人们一边享受着现有科技产品给生活带来的便利，一边畅想着更贴心的产品，带来更便利的生活。各种关于人工智能的科幻电影也让人们对各类智能机器服务人类充满了幻想。还有企业推出的各种智能产品，在媒体的夸大宣传下，也使得人们的期望越来越高。2011 年由 IBM 设计的 Watson 在关于问答比赛的节目《危险边缘》中战胜了人类选手（见图1.1），这让人们觉得计算机可以拥有“智慧”，能“思考”，甚至很多人开始担心人类社会会毁于人工智能。不管相不相信智能机器人和智能应用的潜在危机，人们已经离不开生活中的各种智能应用和智能机器人所提供的服务，并且期望它们能够提供更多更智能的服务。

William Wilkinson's "An Account of the Principalities of Wallachia and Moldavia" inspired this author's most famous novel.



图 1.1: Watson 在关于问答比赛的节目《危险边缘》中战胜了人类选手

智能搜索引擎所服务的人群广泛，这项技术近几年已经有了稳步发展，如国内某著名搜索引擎，图1.2展示了在该搜索引擎¹上查询“姚明的身高是多少”的查询结果，结果页面的第一条就直接展示了确切的答案，而答案的来源出自“姚明”的百科页面²，在该页面的结构化数据栏中我们找到了答案的来源（见图1.2）。从这个例子中我们可以看出智能搜索引擎现在已经能够理解一些简单的查询，并利用已有知识库（如百科页面中的结构化数据栏中的知识），直接而准确地回答这些问题，但也仅限于简单的情况，对于复杂的例子，现阶段的搜索引擎也无法理解用户的查询意图，只能使用传统的手段，把相关的页面作为结果返回。要想提供更好的服务，则需要更高更成熟的技术支持。

¹<https://www.sogou.com/>

²<https://baike.sogou.com/v4957112.htm?fromTitle=%E5%A7%9A%E6%98%8E>



图 1.2: 在某搜索引擎上查询“姚明的身高是多少”的查询结果

中文名	姚明	外文名	Yao Ming
别 名	明王、移动长城、小巨人、大姚	国 籍	中国
民 族	汉族	出生地	上海市徐汇区
出生日期	1980年9月12日	毕业院校	上海交通大学安泰经济与管理学院
身 高	229厘米	体 重	140.6公斤
运动项目	篮球	所属运动队	已退役

图 1.3: “姚明”的百科页面中的结构化数据栏

生活辅助型智能应用也越来越多，给人们生活带来了非常大的便利。人们一直希望生活中有很多如助手般的应用来辅助我们的生活，替我们完成一些机械而繁琐的操作，如使用 Siri 来帮我们查一下关心的球队的赛程情况（见图1.4），只需用自然语言以语音的形式下达命令，应用就能够理解命令，并帮我们完成所有查询操作，直接返回准确的结果。人们也希望有更多更方便的生活辅助型智能应用来服务我们的生活。

智能机器人也开始逐步进入市场，为人们提供一些简单的服务，如聊天机器人，服务员机器人等，未来智能机器人的应用将会更广泛，人们在生活中的各方面都想要有智能机器人来提供服务。如现在的空巢老人问题十分严重，老人需要有人看护和陪同，人们希望有智能机器人可以为老人做健康管理，及时汇报老人的身体状况，还能帮老人做一些简单的家务，与老人有更多的互动交流。能够根据人们需求完成家务的智能服务机器人，将帮助人们从繁琐的家务劳动中解脱出来。

自然语言理解是机器人和应用（计算机）能够提供智能服务的关键技术之一。由于计算机所做出的操作和反应都是基于设计好的程序语言，程序语言属于形式化语言，具有严格定义的形式和语法。人类若想指挥计算机来提供服务，需要给计算机输入程序化的指令，这需要一定的专业知识或者特制的辅助输入界面，这并不是“智能”的做法。计算机只有能够理解自然语言，才能真正明白人类所提出的问题和指令到底是什么，进而给出准确的答复或进行正确的操作。

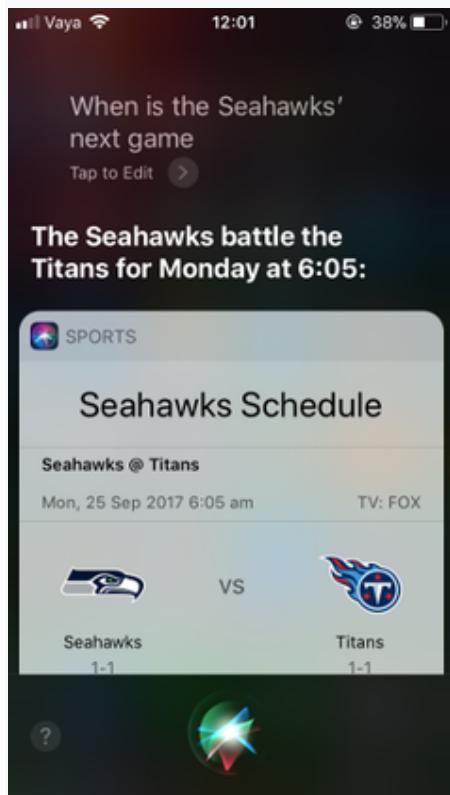


图 1.4: 使用 Siri 查询球队的下一场比赛情况

在学术方面，自然语言处理越来越受关注和重视。自然语言处理是利用计算机对人类的自然语言的信息进行处理和加工的技术 [159]。一般的，自然语言处理研究的问题可以分为以下几个层次：形态学（morphology），语法学（syntax），语义学（semantics）和语用学（pragmatics），如图1.5所示，问题的难度随着层级的升高也越来越大。其中语义学研究的重点是语言单元表达了什么意思。语义解析属于语义学的研究范畴。

语义解析是自然语言处理中的核心任务之一 [110]，也是实现自然语言理解的关键，前面所介绍的智能搜索引擎，智能应用和智能机器人，要想提供更高水平的服务，都离不开语义解析技术的支持。语义解析技术能把自然语言转换为计算机语言，从而使得计算机真正理解自然语言。但现阶段的语义解析技术并不成熟，前面所介绍的一些“理解”的情形，往往是在特定的情境下，依靠一些特定的模板和规则来实现的，目前的语义解析技术还不能做到开放域和无约束条件。

语义解析对自然语言处理领域其他任务的发展也具有积极的推进作用，如机器翻译、自动问答、信息抽取、情感分析等。机器翻译是用计算机将一种自然语言（源语言）自动翻译成另一种自然语言（目标语言）[160]。研究人员先后尝试过使用词对齐（基于词的翻译模型 [20]），短语对齐（基于短语的翻译模型 [138]），句法树映射（基于句法树的模型 [155]），编码器-解码器（神经网络机器翻译 [69]），所使用到的资源越来越高层次（字-短语-句法-语义），翻译效果也是越来越好。如果我们能够

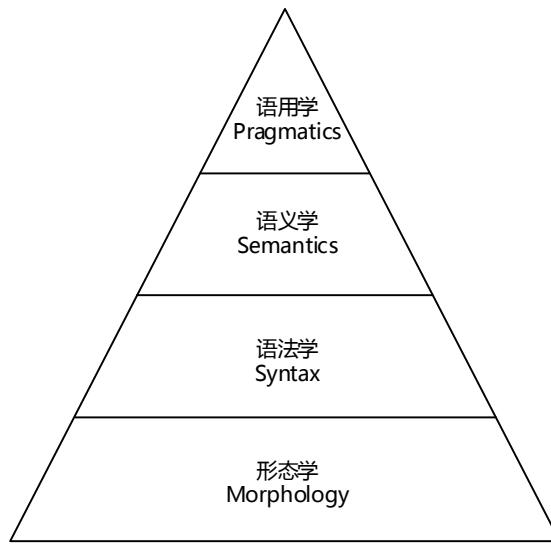


图 1.5: 自然语言处理所研究问题的不同层次

得到源语言的正确的语义表示，将有助于获得理想的翻译效果。同理，如果我们能够利用语义解析得到句子的语义表示，那么我们获得问题的答案 [59, 48, 44, 145]、从句子中抽取知识 [28, 113]、或者判断句子的情感都将更加准确。Corona et al. [37]还提出使用语义解析技术来提高语音识别的准确度。

1.2 语义解析任务介绍

语义解析是指将自然语言句子解析成完整的，机器可识别的或可计算的语义表示（如逻辑表达式）[133]，一旦能够得到句子的语义表示，机器可以对此执行（execute），并给出相应的答案和操作 [93]。图1.6给出了一个语义解析的例子，例子中，语义解析把句子“奥巴马出生在哪个城市？”解析成逻辑表达式 $\lambda x. \text{城市}(x) \wedge \text{出生地}(\text{奥巴马}, x)$ ，该逻辑表达式是一种形式化的表示，能够被机器识别，它表示需要找到变量 x ，并满足 x 的类别是城市，并且“奥巴马”与 x 之间存在出生于的关系，在知识库的支持下机器可以直接执行该逻辑表达式，求得变量 x 是“火奴鲁鲁”。

狭义的语义解析一般只包括解析这一部分，并不涉及后面的执行（如上面的例子），但往往执行与解析是相关的，解析的目的就是为了执行。因此为了便于执行，一般语义解析使用的是执行过程中用到的知识体系。如图1.6中的例子，执行部分所使用到的知识体系是知识库，知识库里面是三元组的知识，对应的语义表示也是基于这种形式构建的。另外，对语义表示执行后的结果一般可以作为有效信息来监督语义解析模型的学习。在本文中，我们认为执行也是语义解析的重要组成部分。

语义解析在解析和执行中都需要知识体系的支撑，根据知识体系的不同构建

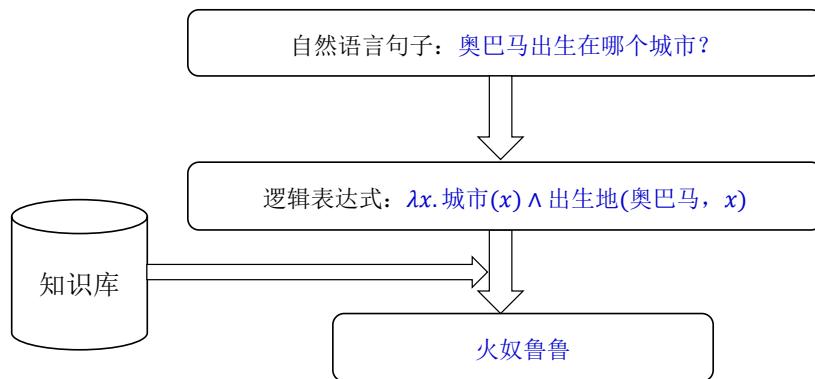


图 1.6: 语义解析的一个例子

情境，语义解析主要可以分为如下两类：

- 基于知识的语义解析：图1.6中的例子就属于这种情境，这种情境一般以知识库为知识体系。解析过程中需要使用知识库中的知识本体作为语义表示的基础框架，执行过程中需要使用知识库中的知识，给出确切的答案。执行过程不需要与知识库以外的其他资源交互。
- 面向机器控制的语义解析：这种情境的执行是具体的机器指令，如在 Linux 系统中使用自然语言对文件进行相关操作，语义解析得到的语义表示就是 Bash 指令，这种情境下就以机器指令体系为语义解析的知识体系。解析过程需要使用机器指令体系作为语义表示的基础框架，执行过程中一般需要与外部世界进行交互，如指挥机器人帮忙倒一杯茶。

在本文中我们主要针对第一类语义解析展开研究。

知识库在这类语义解析中起到了重要的作用：一是得到句子的语义表示之后，利用知识库中的知识可以得到答案；二是知识库的体系是构建语义表示的基础框架。现阶段已存在各种大规模知识库，如 Freebase³、Dbpedia⁴等，图1.7展示了 Freebase 中关于实体 “Barack Obama” 的相关知识，在图1.6的例子中，变量 x 对应的值就是图1.7中的 “Honolulu”。同样的例子，语义表示 $\lambda x. \text{城市}(x) \wedge \text{出生地}(\text{奥巴马}, x)$ 中，“城市” 对应到知识库中的一元谓词，即图1.7中的 “Type-City”，关系 “出生于” 对应到知识库中的二元谓词，即图1.7中的 “PlaceOfBirth”。

句子的语义离不开句子中词语的语义，也离不开句子的结构。语义解析就是研究如何获取句子中词语的语义，如何获取句子的结构，并如何基于词语的语义，利用句子的结构组建句子的语义表示。

³Freebase 现已关闭，现以 Wikidata 的形式呈现

⁴<http://wiki.dbpedia.org/>

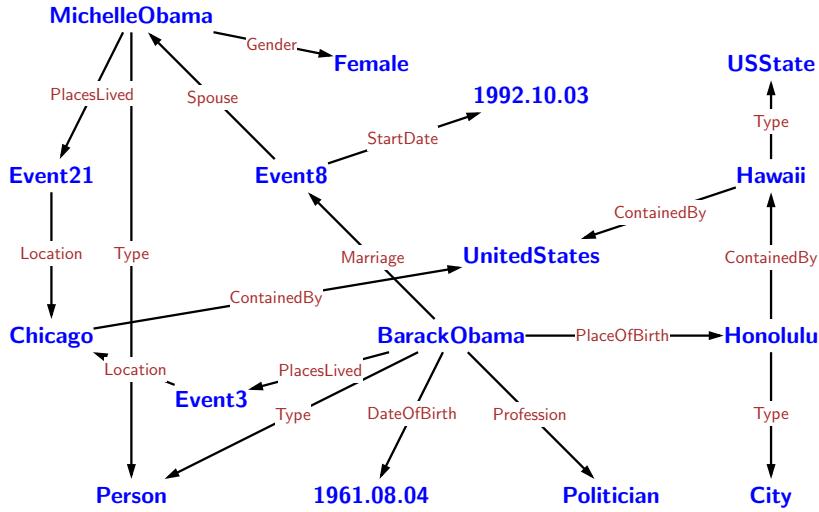


图 1.7: Freebase 中关于实体 “Barack Obama” 的相关知识

1.3 本文的主要研究内容及组织结构

完成语义解析主要需要两个功能模块,一个是语义落地(Semantic Grounding),另一个是结构预测(Structure Prediction)。本文紧紧围绕上述两个模块展开研究,在语义落地方面,由于词典一般需要学习,在开放域的情况下存在覆盖度不足的问题,针对该问题,本文开展了基于词典扩充学习的语义解析方法研究。在结构预测方面,针对句子结构与目标语义表示之间的结构不匹配问题,本文开展了基于句子重写的语义解析方法研究。最后,本文综合利用语义图的语义表示优势和神经网络模型的强预测能力,提出了一种新的语义解析框架——基于端到端的语义图生成的语义解析方法。

围绕上述内容,本文共分为六章,整体组织结构如图1.8所示。本章主要介绍研究背景与意义,以及研究的主要内容。后续章节内容安排如下:

- 第二章首先介绍语义解析的相关背景知识,以及研究语义解析所要了解的一些概念,包括语义的表示,知识库本体,解析算法,学习算法,以及常用的公开数据集。接着对常见的语义解析方法进行分类,并简要介绍其中代表性方法及其优缺点。最后简要介绍几种基于语义解析的典型应用。
- 在第三章,针对语义解析中用于语义落地的词典往往存在覆盖度不够的问题,我们先后提出了两种基于词典扩展学习的语义解析方法。一种是在现有工作的基础上,在首轮训练的过程中自动引入新的词汇,然后在后续的训练中对新词汇进行打分,最后再利用投票机制在所有词汇中选择核心词汇组成核心词典,并应用于语义解析中。另一种是充分利用容易获取大规模的文本语料和词典资源来学习词语与词语之间的语义相似性度量,然后在基于图的半监

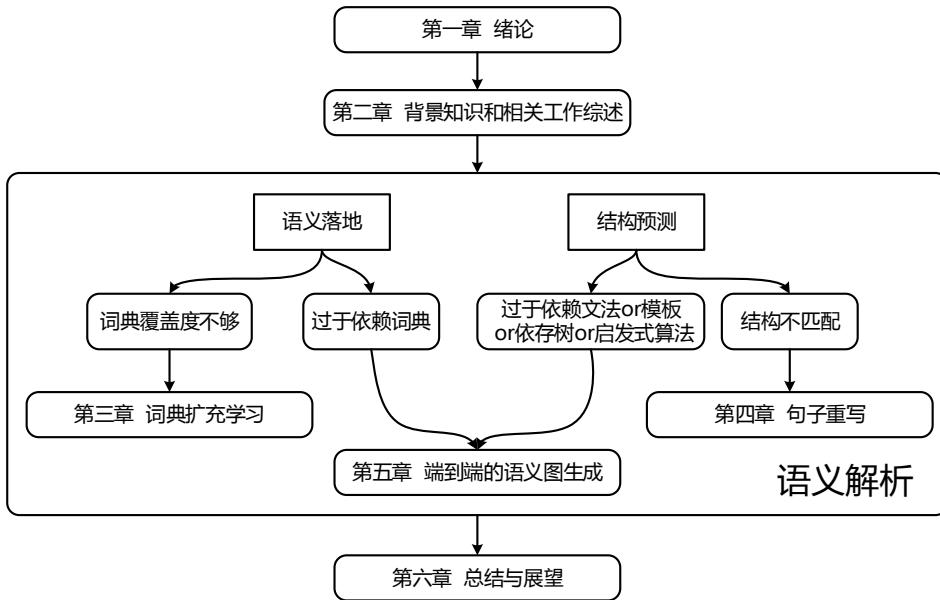


图 1.8: 本文的组织结构

督学习的框架里，从少量种子词汇出发，学习出新的词汇。该方法的出发点是相似的词语应该映射到相同的谓词。

- 在第四章，针对语义解析中结构预测时，句子与目标语义表示之间的结构不匹配问题，我们提出了一种新的语义解析框架——基于句子重写的语义解析方法。该方法的出发点是一个句子总存在另一种表述形式（新的句子），该表述形式与原句子的意思一样，同时其结构与目标语义表示的结构是匹配的。该章对结构不匹配的问题进行分析，并总结出两类结构不匹配的问题，分别是 1-N 型不匹配和 N-1 型不匹配，进而分别提出基于字典替换的句子重写方法和基于模板替换的句子重写方法。最后定义一个联合模型，综合考虑句子重写部分和语义解析部分。
- 在第五章，针对语义解析中语义落地部分过于依赖于词典，以及结构预测部分依赖于设计文法，模板和启发式的规则的综合问题，我们提出了一种基于端到端的语义图生成的语义解析方法——Sequence-to-Action。该方法能够同时利用了语义图的语义表示优势，以及循环神经网络模型的强表示学习能力和序列预测能力。基于该方法能够把语义解析过程转化为语义图生成过程，同时定义动作集来编码语义图的生成过程，从而可以把语义解析看作句子的词语序列到动作序列的翻译过程，进而利用循环神经网络模型来建模动作序列的生成。此外，在解码的过程中句法约束条件和语义约束条件能够方便地加入，进一步提高语义解析的准确度。
- 第六章是对全文工作的总结以及对下一步研究工作的展望。

第二章 语义解析方法综述

2.1 概述

语义解析是将自然语言句子解析成计算机可识别或可计算的语义表示（如逻辑表达式）的过程 [133]。语义解析是自然语言处理领域基础性任务之一，也是实现自然语言理解所需的技术。

语义解析已经有很久的研究历史，近年来更是吸引了大批的研究者。早年的语义解析都是基于规则的，直到 90 年代，机器学习开始兴起，各种机器学习算法开始运用于语义解析。近年来，随着深度学习的兴起，也涌现了很多基于深度学习的语义解析方法。

本章，我们将首先介绍语义解析的相关背景知识，接着介绍研究语义解析之前要了解的相关概念，接着我们对语义解析方法进行分类和总结，最后介绍若干语义解析的应用。

2.2 组合语义学和模型理论语义学

现阶段大部分语义解析方法都是基于组合语义，组合语义的基本准则是：一个句子的整体意思是句子中各个成分意思的组合。该理论最早由哲学家 Richard Montague 提出，这种语义学理论被称为蒙太古语义学 [108, 107, 109]。

模型理论语义学 (model-theoretic semantics) 是基于模型理论的语义学，其中模型理论 (model theory)¹是出自数理逻辑中的概念，广义而言，模型理论是对语言的解释，这里的语言可以是人类的自然语言，也可以是形式化语言。在自然语言中，基于模型理论的语义学指的是，用世界中的某一个状态 (state) 来对一个句子进行解释，图2.1展示了一个例子。

Block 2 is blue.

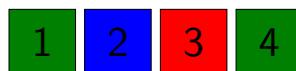


图 2.1: 用世界中的状态来解释句子的例子

现阶段的语义解析方法大部分都基于这两个理论：组合语义和模型理论语义，前者强调语义是组合而成的，具有结构性，将语义解析看作是一个结构预测 (Structure Prediction) 的问题。后者强调句子的语义对应到物理世界中的某个状态，具体

¹<https://plato.stanford.edu/entries/model-theory/>

实践中就是对应到一个知识体系中，句子中的实体和关系都用知识体系中的实体和关系来解释，这也意味着对一个自然语言句子进行语义解析需要把句子中的实体和关系映射到知识体系中的实体和关系，这是一个语义落地（Semantic Grounding）的过程。综上所述一个语义解析器通常需要两个功能：语义落地和结构预测。

2.2.1 语义落地

语义落地是指句子中的实体对应到物理世界（world）中的实体，句子中的关系对应到物理世界中的关系 [78]。一般的，我们使用知识库中的知识来表示一个物理世界，即把句子中的实体映射到知识库中的实体，把句子中的关系映射到知识库中的关系（也称谓词，包括一元谓词和二元谓词）。

语义解析过程中语义落地的实现方式一般有两种，一种是利用词典。词典是词汇的集合，词汇定义了句子中的词语到知识库中实体谓词的映射，这些词汇一般需要提前学习，或者在训练的时候利用解析器来学习。不管是哪种词典学习方式，最终在解析新句子的时候，词典一般是固定的，这样以来只有词典覆盖到的知识库中实体和谓词的映射才能被触发（trigger），也就是有了映射（词汇）才能利用这个映射（词汇触发）。利用词典的语义解析一般都是基于组合文法的方法。另一种实现语义落地的方式是语义匹配。这种方式一般不需要提前学习词语到谓词的映射，而是根据解析过程的需要来触发的。这种语义解析方法通常是先进行结构预测，再进行语义落地。这种情况下，语义解析先得到句子的一个结构化表示，一般使用图来表示，图中的节点表示实体或者变量或者类别，图中的边表示关系，这时这些实体、变量、类别和边都是没有语义落地的，随后通过语义落地把图中的节点和边都映射到知识库中的项。具体的，一般先进行实体的映射，这样做有两个好处：一是相比先映射关系，利用上下文来映射实体能够更加的精准；二是先进行实体的语义落地，紧接着对关系进行语义落地时，落地的对象就能够有所限制，必须是与该实体相关的关系谓词。实体的语义落地之后再进行关系的语义落地，这需要考虑可落地的关系的全部候选，利用打分函数从中选择最好的。与前面利用词典的方法不同，这种方法不需要提前知道词语到谓词的映射，而是考虑所有可能的映射，前者容易因为事先缺失某种映射而无法完成解析，后者就不存在这种情况；但是后者的搜索空间一般会很大，而前者的搜索空间完全由词典的大小来控制。

在研究语义落地的过程中，衍生出如下几个研究点：词典学习，词典扩充和知识库谓词扩充。

2.2.1.1 词典学习

若使用词典来实现语义落地，词典的质量和覆盖度一定程度上决定了语义解析的性能。因此词典学习是语义解析中一个重要的问题 [77]。词典学习中最具有代

Words	Category
	Syntax : Semantics
Texas	NP : <i>texas</i>
borders	(S\NP) /NP : $\lambda x. \lambda y. borders(y, x)$
Kansas	NP : <i>kansas</i>
Kansas city	NP : <i>kansas_city_MO</i>

图 2.2: CCG 中的词汇的例子

Input Training Example	
Sentence:	Texas borders Kansas
Logic Form:	$borders(texas, kansas)$
Output Lexicon	
Output Substrings:	Output Categories:
Texas	NP : <i>texas</i>
borders	NP : <i>kansas</i>
Kansas	(S\NP) /NP :
Texas borders	$\lambda x. \lambda y. borders(y, x)$
borders Kansas	
Texas borders Kansas	

图 2.3: GENLEX 中生成候选词汇的图例说明

表性的工作有限定领域下 Zettlemoyer and Collins [154]提出的词典学习方法——GENLEX，和开放领域下 Cai and Yates [22]提出的词典学习方法——LEXTENDER。

GENLEX 方法 Zettlemoyer and Collins [154]利用组合范畴文法 (Combinatory Categorial Grammar, CCG)²来进行语义解析。CCG 的核心就是词典，CCG 中词汇的语义表示端由两部分组成，一部分是句法信息，另一部分是语义信息。图2.2展示了 CCG 中的几个词汇。他们的词典学习方法分两步：首先从样例中生成所有可能的词汇的候选，然后利用这些候选的词汇（再加上当前模型的词典）对样例进行解析，再把在正确的解析中所用到的候选词汇加入当前模型的词典。

第一步的具体操作是对句子端取句子中所有可能的短语，对逻辑表达式端利用规则抽取所有可能的语义项，然后把所有短语和所有语义项进行交叉，短语和语义项的一个配对构成一个词汇，这个过程如图2.3所示。

第二步相当于对第一步所生成的候选词汇进行筛选。第一步候选词汇生成的准则是：不漏掉任何一个词汇的可能性。这种形式带来的影响就是候选的词汇太多，因此需要对候选词汇进行筛选。词汇筛选的思想也很简单：对样例进行解析后，能够得到正确解析结果的解析过程中所使用到的词汇就是正确的词汇，把这些词汇收集起来就能构成语义解析所需要的词典。

²在后面我们会直接使用 CCG，而不使用中文说明。

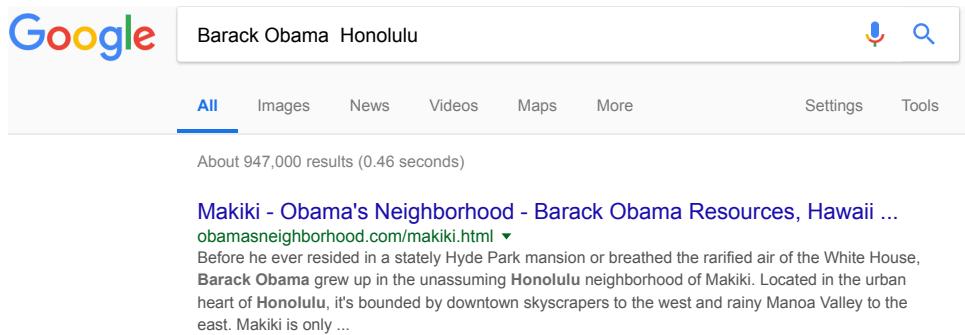


图 2.4: 利用 Google 搜索 “Barack Obama” 和 “Honolulu” 得到的结果

LEXTENDER 方法 Cai and Yates [22]利用额外语料来学习词典。这个方法是基于模板匹配的，具体的，从知识库本体中构建知识库模板，再利用搜索引擎的结果页面来构建文本模板。实际上，知识库模板就是知识库中的一个模式(schema)³，每个 schema 下都有很多实例，如 Place_of_birth 关系下面有很多实例（满足这个关系的实体对），如 “Barack Obama” 和 “Honolulu”。当把这两个实体输入到搜索引擎的时候，如 Google⁴，在查询结果中就会有句子包含这两个实体（如图2.4所示）。基于这样的假设：自然语言句子中两个实体之间词语可能是这两个实体之间的关系；若这两个实体是某个谓词下的一个实例，就称该词语与该谓词有一次实体对共现，实体对共现的次数越多，表明该词语越可能映射到该谓词。于是从例子中，就可以得到这样的可能的词汇：grew up in :: Place_of_birth。

2.2.1.2 词典扩充

然而仅仅利用训练数据，以及知识库与文本语料之间的实体对共现来学习词典，仍会存在词典覆盖度不够的问题。为了解决这个问题，需要进行词典的扩展学习。通常，可以使用半监督学习方法，利用文本与文本之间的语义相似度，从少量种子词汇中学习新的词汇以扩充词典，这个工作我们将在第三章进行详细的说明。

2.2.1.3 知识库谓词扩充

语义落地没能顺利完成可能有两个原因，一个是词典的覆盖度不够，就是知识库端存在该谓词，但是句子端的词语无法映射过去。另外一个是知识库不完备，就是知识库端不存在能够表示该句子中某个关系的谓词。前者可以用词典学习和词典扩充学习来解决，后者需要对知识库进行扩充。为此，Krishnamurthy and Mitchell [81]提出利用抽取出来的三元组来充当知识库，Gardner and Krishnamurthy [52]提

³在后文中直接使用 schema，而不加中文说明

⁴<https://www.google.com/>

出同时利用知识库的知识和抽取出来的半结构化的知识，相当于扩充了知识库中的谓词。

2.2.1.4 基于神经网络模型的语义匹配

前面已经提到语义落地有两种方法，一种是利用词典的方法，另一种是利用语义匹配的方法。后者需要对所有可能的候选进行排序打分，为了更好的建模这种与语义相关的映射，最近越来越多的研究者倾向于使用神经网络模型。其中最典型的是 Yih et al. [149] 使用卷积神经网络（CNN）模型来建模这种映射。

2.2.2 结构预测

语义解析中结构预测指的是对句子所对应的语义表示的结构的预测。通常结构的形式有两种，一种是基于逻辑表达式的，另一种是基于图的。这两种形式的表达是等价的，两者之间可以按照一些特定规则进行相互转换。使用基于逻辑表达式来表示目标语义的结构的方法一般是基于组合规则将小的语义单元组合成更大的语义单元。组合规则的设定和语义图的构建方法是语义解析结构预测研究的重点。

2.2.2.1 组合规则

利用组合规则来进行结构预测的方法，通常与词典一起使用，词典负责底层的词语的语义落地，组合规则负责把底层的语义表示单元逐层往上组合成更大的语义单元。组合规则一般是从语言现象中抽象出来的规则。如组合范畴文法（CCG）[127, 128] 中一般使用四条组合规则就可以覆盖大部分的语言现象。这四条组合规则分为两组，分别在图2.5和图2.6中展示：

$$\begin{array}{lll} A/B : f & B : g & \Rightarrow A : f(g) \quad (>) \\ B : g & A \setminus B : f & \Rightarrow A : f(g) \quad (<) \end{array}$$

图 2.5: CCG 中的函数应用 (functional application) 规则，包括前向和后向两种

$$\begin{array}{lll} A/B : f & B/C : g & \Rightarrow A/C : \lambda x. f(g(x)) \quad (> \mathbf{B}) \\ B \setminus C : g & A \setminus B : f & \Rightarrow A \setminus C : \lambda x. f(g(x)) \quad (< \mathbf{B}) \end{array}$$

图 2.6: CCG 中的函数组合 (functional application) 规则，包括前向和后向两种

图中的规则形式是用于语义解析的形式，最早由 Zettlemoyer and Collins [154] 提出，规则中的项（如， $A/B : f$ ）都包含两部分，前面部分表示句法（如， A/B ），后面部分表示语义（如， f ），去除所有语义部分的规则就是 CCG 中规则的最原始的形式。 A/B 表示一个范畴 (category)，当该范畴的右侧（“/”表示右侧缺失）补全一个类型是 B 的范畴的时候，该范畴的类型是 A 。按照规则， A/B 和 B 可以

进行组合，得到类型是 A 的范畴。语义解析中函数应用的两种形式的规则使用最频繁，图2.7展示了基于函数应用规则进行语义组合的例子。

$$\begin{array}{c}
 \text{Utah} \qquad \text{borders} \qquad \text{Idaho} \\
 \hline
 NP \qquad \frac{(S \setminus NP)/NP}{\lambda x. \lambda y. \text{borders}(y, x)} \qquad NP \\
 utah \qquad \xrightarrow{\qquad \qquad \qquad idaho \qquad \qquad} \\
 \qquad \qquad \qquad \frac{(S \setminus NP)}{\lambda y. \text{borders}(y, idaho)} \\
 \hline
 S \\
 \text{borders}(utah, idaho)
 \end{array}$$

图 2.7: 利用规则进行语义组合的例子

CCG 中还有其他规则，篇幅所限这里我们不再一一介绍，其实每一个规则都是从语言现象中抽象出来的组合规则，每一个特定的词类，都有固定的表示，如一个及物动词，它的句法部分的表示为: $(S \setminus NP)/NP$ ，具体来讲，它需要右侧补全一个宾语（对应 NP ），还需要左侧补全一个主语（对应 NP ），三者利用规则组合以后，得到类型为 S 的范畴，即一个结构完整的句子。当组合文法的粒度很小的时候，会使得组合起来的中间单元过多，Zhang et al. [156]提出使用粒度更大的宏观规则，来提升解析的效率。

2.2.2.2 语义图

相比语义解析，句法分析和依存分析技术要更加成熟，直接在句法分析或依存分析的基础上进行语义解析不失为一个好策略。把句法分析或依存分析的结果（树结构）作为语义表示的框架，这与利用类似于 CCG 的方法不同的是：句法和语义部分是分离的，先做句法，再做语义，而使用类似于 CCG 的方法中句法和语义是同步进行的。这种方法的出发点是：句子的句法结构和语义结构大部分情况下是同构的。使用语义图来作为语义表示的结构的方法的优势是：不需要使用额外的组合规则，只需要使用现成的句法分析器或者依存分析器就可以得到句子的树结构。而不足的地方是句子的句法结构与语义结构可能存在异构的情况，这时候需要对句法结构进行调整才能做语义部分。

语义图的构建方法除了使用句法分析和依存分析以外，还可以使用模板和动态构建方法，在后续章节我们会进行详细介绍。

2.3 语义解析

前面介绍了语义解析的一些相关背景知识，这一节我们将介绍在实际进行语义解析研究之前还要了解的几个要素。

2.3.1 语义表示

语义解析是将自然语言句子解析成计算机可识别或可计算的语义表示的过程。我们首先要定义语义表示的形式。一个好的语义表示应该具有如下要素：

- 机器友好的：由于该语义表示是要输入给机器使用的，相当于机器指令，所以语义表示最好是类似于程序语言，或者方便转化为程序语言的形式。
- 人类友好的：为了方便研究，语义表示的形式应该是人类可以看懂的，或者是能够方便转化为人类容易看懂的形式。
- 自然语言友好的：由于需要把自然语言句子映射到该语义表示，为了降低这个映射过程的难度，应该让语义表示与自然语言拥有尽可能多的共同点，比如：结构类似。
- 表示能力强的：由于该语义表示是用来表示自然语言的意思（meaning）的，那么该语义表示应该具备完备性，即每一个自然语言句子，都应该能用该语义表示表示出来。

接下来，我们将介绍现阶段比较流行的语义表示，并分析他们用于语义解析时的优点和不足。

2.3.1.1 基于向量的语义表示

近年来，深度学习在自然语言处理领域广泛应用。研究者不仅用向量来表示词，还用向量来表示句子，甚至段落和篇章。从词的向量表示到句子的向量表示，其中的组合方式也是多种多样，有直接使用加法的，使用乘法的，使用矩阵映射的，使用张量的，使用递归神经网络的，等等。Krishnamurthy and Mitchell [80]提出一种通用的基于向量空间的语义解析框架。一旦获得句子或篇章的向量表示，研究者可以用该向量表示做下一步工作，如分类，包括情感分类，文本分类等；如序列生成，包括机器翻译，语义角色标注等。

基于向量的语义表示方法的优点有：a) 非常通用的一种表示，可应用于各种后续的处理，对于机器学习任务也是非常友好的；b) 不再需要设计特征。

基于向量的语义表示方法的缺点有：a) 该表示只是整体的一个近似表示，很难区分精细的语义差别，这对于追求完全准确表示的语义解析来说略显粗糙；b) 从词的向量化表示再到句子的向量化表示的组合过程，很难建模，很难找到一个有效的方式来捕捉词与词之间的依赖关系；c) 对人类不友好，就像黑盒子（black box）一样，人类很难从向量表示中看出具体语义。

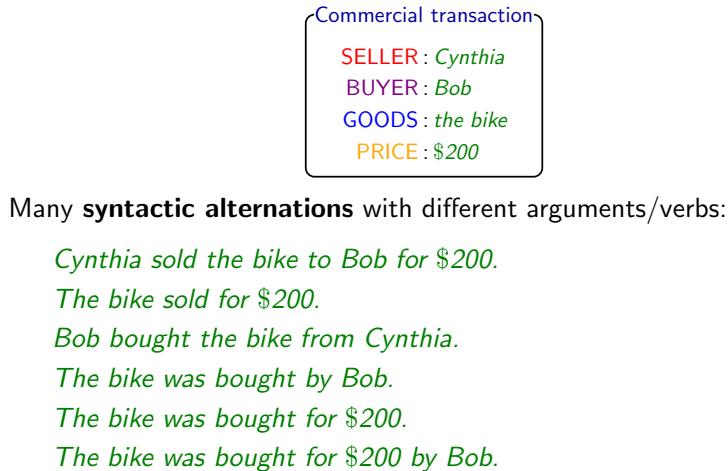


图 2.8: 句子结构不同, 但是基于 FrameNet 的语义表示相同的例子

2.3.1.2 基于框架的语义表示

Minsky [105]首先提出利用框架来表示语义, 他对框架的定义是: 一个框架是用于表示原型情境的数据结构, 如一个小孩的生日派对。一个框架有一个谓词 (predicate) 表征该框架, 一个框架对应一个抽象化的情境, 此外, 一个框架还有若干槽 (slot), 也可以成为论元, 通常具有一些语义的角色, 用来表示与该框架有关的额外的信息。这种谓词-论元的表示结构起源 Baker [8]与 Fillmore [50]提出的格文法 (case grammar)。基于上述理论, 有两个基于框架的语义表示的具体的实现, 分别是 FrameNet⁵和 PropBank⁶。

FrameNet[9] 是 Berkeley 大学建立的一个工程, 最早公布于 1998 年。该工程的主要特征有: a) 为语义和句法普遍化提供语料证据; b) 对目标词语 (主要是名词、形容词、动词) 的价位 (valence) 表示, 其中语义部分主要利用了框架语义学。这个工程的主要成果是提供了一个数据库, 数据库主要包括如下三部分: 词典、框架数据库和基于词典和框架标注好的句子集。FrameNet 的出现有力地推进了语言学和自然语言处理的发展, 还直接导致了语义角色标注任务的诞生。

FrameNet 以框架为核心, 辅助以论元的标签, 并且论元的标签在不同的框架里是可以共享的, 如 SELLER 这个论元同时出现在 SELL 和 BUY 两个框架中。FrameNet 通过对不同形式的词汇单元进行标准化, 使得句子的基于框架的语义表示是可以不依赖于句法结构, 即只要句意相同, 它们基于 FrameNet 中的框架的语义表示也是相同的, 图2.8展示了这样的一个例子⁷。

PropBank[75, 112] 是在宾州树库 (Penn Treebank) 的基础上, 对这个树库

⁵<https://framenet.icsi.berkeley.edu/fndrupal/>

⁶<https://propbank.github.io/>

⁷例子借用于 Percy Liang 于 2015 年在 ICML 做的报告: “Natural language understanding: foundations and state-of-the art”。

中动词的配价关系的进行标注，而建立的“命题树库”（Proposition Bank，简称 PropBank）。与 FrameNet 相似，PropBank 标注的也是谓词-论元结构信息。不同的是，PropBank 是以动词为核心，并基于句法结构，对论元信息进行标注，论元标签对于动词来说是特定的，即一个动词，对应一组论元标签。图2.8中基于 FrameNet 的语义表示是相同的，但是基于 PropBank 的语义表示就会出现多种形式。从这个角度来讲，FrameNet 比 PropBank 对于语言现象具有更好的描述能力。然而，PropBank 是基于 Penn Treebank 构建的，数据规模更大，从而也吸引了大批学者使用 PropBank 进行相关研究。

基于框架的语义表示方法的优点有：a) 具有很好的结构性，与自然语言句子的结构类似；b) 具有很好的可理解性，研究者可通过框架理解该框架所对应的情境，如该情境中的施事者，受事者分别是什么；c) 对机器友好，结构化的表示，很方便转换成机器可识别的程序语言；d) 具有坚实的语言学理论基础。

基于框架的语义表示方法的缺点有：a) 需要人类来设计这些框架，一个框架对应一个情境，很难做到框架的覆盖度能够覆盖到表示世界（world）中的所有情境；b) 框架的粗细程度设计很难把握，框架的粒度太大，导致无法进行细致区分，框架的粒度太小，导致框架的数量太多。

近年来，有研究者利用框架来表示句子语义，这类语义解析称为 Frame-Semantic Parsing，是将一个句子映射成基于框架的语义表示的过程 [40, 39, 41]。

2.3.1.3 基于逻辑表达式的语义表示

lambda-表达式 (lambda-calculus) 是一种基于数理逻辑理论的形式化的语义表示体系。它可以作为一种粘合语言 (glue language) 来组合自然语言中的词语，图2.9展示了几个句子与其对应的 lambda-表达式的例子。具体的，lambda-表达式可以通过两个函数来进行组合。这两个函数分别是函数应用 (function application) 和函数组合 (function composition)。函数应用 (function application) 指的是将一个函数 $\lambda x.f(x)$ 应用于具体的一个参数 a ，具体形式如下：

$$\lambda x.f(x) \quad a \rightarrow f(a)$$

函数组合 (function composition) 指的是将一个主要 (primary) 的函数 $\lambda x.f(x)$ 应用于一个次要 (secondary) 的函数的结果上 $\lambda y.g(y)$ ，具体形式如下：

$$\lambda x.f(x) \quad \lambda y.g(y) \rightarrow \lambda y.f(g(y))$$

lambda-DCS 树 [91] 是基于 DCS (dependency-based compositional semantics) 而构建的形式类似于 lambda-表达式的语义表示形式。相比于 lambda-表达

- a) What states border Texas
 $\lambda x.state(x) \wedge borders(x, \text{texas})$
- b) What is the largest state
 $\arg \max(\lambda x.state(x), \lambda x.size(x))$
- c) What states border the state that borders the most states
 $\lambda x.state(x) \wedge borders(x, \arg \max(\lambda y.state(y), \lambda y.count(\lambda z.state(z) \wedge borders(y, z))))$

图 2.9: 句子及其对应的基于 lambda-表达式的语义表示的例子

- Utterance: “*people who have lived in Seattle*”
- Logical form (lambda calculus): $\lambda x.\exists e.\text{PlacesLived}(x, e) \wedge \text{Location}(e, \text{Seattle})$
- Logical form (lambda DCS): `PlacesLived.Location.Seattle`

图 2.10: lambda-表达式和 lambda-DCS 对于一个句子的表示的对比

式, lambda-DCS 的表示形式更加的简单, 图2.10展示了一个对比例子。DCS 是基于严格的树结构的语义表示形式, 显然它的表示能力受限于树结构, 相比 lambda-表达式, DCS 的语义表示能力要弱一些, 对语言中的某些现象不能进行表示, 如指代。lambda-DCS 是借用了 lambda-表达式中的一些思想, 得到的 lambda-DCS 的语义表示形式具有与 lambda-表达式等价的语义表示能力, 同时它还保存了 DCS 本身基于树结构的表示形式, 这也使得这种语义表示形式与自然语言本身更加的贴近。

基于逻辑表达式的语义表示的优点有: a) 表达能力非常强, 基本能够覆盖所有的语言现象; b) 具有丰富的语义, 人类也容易看懂; c) 对机器友好, 该表示本属于形式化语言, 能够很方便, 很确定性的转换为机器可直接使用的程序语言。

基于逻辑表达式的语义表示的缺点有: a) 该表示是领域特定的, 对于一个领域, 要设计该领域中的语言现象和语言知识; b) 很难进行人工标注, 需要一定的专业知识。

2.3.1.4 AMR

Abstract Meaning Representation (AMR) 是一种基于图的语义表示形式 [10]。可以使用 AMR 图来表示一个句子的语义。AMR 图是一个有根节点的、边带标签的、有向无环图。其中节点表示概念 (concepts), 边表示两个概念之间的关系。AMR 中的概念主要如下几种形式: 单词 (如, “boy”), PropBank 中的框架 (如, want-01), 或者特殊的关键词 (包括实体类型, 如 “date-entity”, 和逻辑连接符, 如 `and`)。这种形式是基于命题逻辑 (propositional logic) 和 neo-Davidsonian 事件表示形式 [114, 42] 的, 图2.11展示了句子 “The boy wants to go” 的几种语义表

LOGIC format:

$$\exists w, b, g: \\ \text{instance}(w, \text{want-01}) \wedge \text{instance}(g, \text{go-01}) \wedge \\ \text{instance}(b, \text{boy}) \wedge \text{arg0}(w, b) \wedge \\ \text{arg1}(w, g) \wedge \text{arg0}(g, b)$$
AMR format (based on PENMAN):

```
(w / want-01
  :arg0 (b / boy)
  :arg1 (g / go-01
  :arg0 b))
```

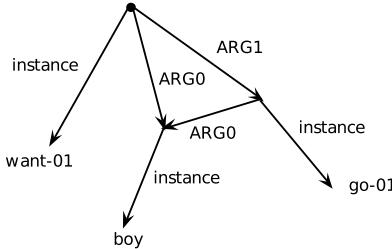
GRAPH format:

图 2.11: 句子 “The boy wants to go” 的几种等价的语义表示

```
(f / fear-01
  :polarity "-"
  :ARG0 (s / soldier)
  :ARG1 (d / die-01
  :ARG1 s))
```

The soldier was not afraid of dying.
 The soldier was not afraid to die.
 The soldier did not fear death.

图 2.12: 几个不同句子拥有相同的 AMR 表示的示例

示，并且这几种语义表示是等价的。在 AMR 中具有相同意思的句子，即使具有不同的句法结构，它们的 AMR 图也是相同的，图2.12展示了这样的一个示例。

使用 AMR 来表示语义的优点有：a) 使用基于图的结构，与句子的依存图很类似，方便获取；b) 对语义进行了一定的抽象，意思表达所覆盖的范围很广；c) 可以建模很复杂的语言现象，如否定，比较等；d) 有大规模语料支持；d) 对人和机器都友好，人容易看懂 AMR 表示，机器也容易对这种语义表示进行处理。

使用 AMR 来表示语义的缺点有：a) 标注比较困难；b) 表示的主体是 PropBank 中的框架，这也使得 AMR 表示受限于 PropBank 中的框架语义表示。

近年来，基于 AMR 的语义分析（称为 AMR 分析）吸引了众多学者 [51, 136, 120, 6, 106]，AMR 分析是将句子映射成 AMR 图的过程。

2.3.2 知识库数据组织形式

对于语义解析，知识库主要有两个作用：知识库本体结构决定了语义表示的结构；知识库中的知识可以用来计算语义表示。知识库数据组织形式是知识库本体结构的具体体现，下面我们简要介绍两种常见的知识库数据组织形式。

三元组 知识库中的数据形式一般以三元组的形式存在，当前众多大规模知识库都采用这种形式，如 Freebase、Dbpedia 等。三元组顾名思义，由三部分组成，对于表示两个实体之间的关系的知识，三元组的形式是：(实体 1, 关系, 实体 2)，关系对应到知识库本体中的二元谓词，如 `PlaceOfBirth`；对于表示实体的类别的知识，三元组的形式是这样的：(实体, 类别, 类别值)，类别值对应到知识库本体中的一元谓词，如 `Person`。

表格 知识库中的数据形式以表格的形式存在，如以 Wikipedia 中的表格作为知识库。表2.1展示了 Wikipedia 中的一个表格。[129]

Year	Competition	Venue	Position	Event	Notes
1975	Asian Championships	Seoul, South Korea	1st	5,000m	14:00.8
1979	Asian Championships	Tokyo, Japan	2nd	10,000m	29:00.2
1981	Asian Championships	Tokyo, Japan	1st	10,000m	28:53.29
1981	Fukuoka Marathon	Fukuoka, Japan	2nd	Marathon	2:09:37
1982	Fukuoka Marathon	Fukuoka, Japan	4th	Marathon	2:11:45
1983	World Championships	Helsinki, Finland	28th	10,000m	29:49.04

表 2.1: Wikipedia 中的一个表格的例子

2.3.3 解析算法

下面介绍几种常见的解析算法。

CYK 基于组合语义的语义解析方法中，最常用的解析算法莫过于 CYK 算法 [32, 70, 150]⁸。这类方法的解析过程是自底向上的，类似于句法分析 (syntactic parsing) 的解析过程，语义解析过程中所使用的文法，如组合范畴文法 (CCG) 和基于依存的组合语义文法 (DCS) 等，就类似于句法分析中的上下文无关文法 (CFG)。语义解析中，最底层的文法，也就是叶子节点的文法，对应到词汇，即词语到谓词的映射；继续往上解析就是利用组合规则进行不断组合更大的单元，直到生成根节点的逻辑表达式。

CYK 算法的最坏的时间复杂度是 $O(n^3|G|)$ ，其中 n 是句子的长度， $|G|$ 是文法的数目。

⁸https://en.wikipedia.org/wiki/CYK_algorithm

基于 agenda 的解析算法 为了进一步提高解析的效率，研究者开始探索其他的解析算法。基于 agenda 的解析算法已经在句法分析中广泛运用 [74, 23, 116]。Berant and Liang [15]最早在语义解析中使用这种解析算法。该算法的效率比 CYK 算法的更高。

2.3.4 学习算法

语义解析器对句子进行解析后会得到很多候选的逻辑表达式，为了从中选择最好的，还需要一个打分函数，同时还需要学习其中的参数。我们以学习算法所使用的监督知识类别为分类标准，将学习算法分为三类：监督学习算法、半监督学习算法、非监督学习算法。

监督学习算法 这种方法所使用的训练数据的形式是（问题，逻辑表达形式）对，通过训练，我们希望得到新句子的逻辑表达式，这需要我们学习到词语到谓词的映射，即词汇，以及组合规则的选择（组合规则一般是定义好的若干条规则）。

半监督学习算法 这种方法所使用的训练数据的形式通常不是（问题，逻辑表达形式）对，可以是（问题，答案）对，这种学习算法以逻辑表达式为中间隐变量，以答案作为远距离监督。虽然监督知识所包含的信息比全监督的方式少，但是利用这些半监督信息依然可以进行有效地学习，而且这种训练数据更容易获取。监督信息还可以是其他形式，Kate and Mooney [71]把 SVM 的分类结果作为一种监督信息来学习语义解析模型；Kate and Mooney [72]中所使用的训练语料是（问题，多个逻辑表达式），其中只有一个逻辑表达式是正确的。另外我们把交互式学习算法也归在半监督学习算法下，交互式学习算法通过用户的一些反馈信息来进行模型的学习，使用这种交互式学习算法的语义解析系统有 Clarke et al. [31], Artzi and Zettlemoyer [3], Wang et al. [137], Iyer et al. [66]。

非监督学习算法 这种方法不需要监督知识，本质上，把语义解析中所要使用到的词语到谓词的映射，即词汇的学习，利用聚类的思想来学习。在语义解析的工作中，使用非监督学习算法的经典工作有：Poon and Domingos [119]、Poon [118]和 Goldwasser et al. [56]。

2.3.5 公开数据集

这里，我们简要介绍目前常用的语义解析公开数据集。

2.3.5.1 GEOQUERY 数据集

GEOQUERY⁹数据集最早在 1996 年由 Zelle and Mooney [151]公布，该数据集是基于美国地理知识的查询数据集¹⁰。该数据集附带有一个关于美国地理知识的小型知识库，知识库中总共有 800 个事实类的知识，知识的表示采用 Prolog¹¹的形式。该知识库主要包括有关美国地理的如下知识：

-
- 州 (state) - 包括州的首府，人口，面积，人口密度，主要的城市，河流以及与之接壤的州
 - 城市 (city) - 包括城市的人口，以及该城市所在的州
 - 河流 (river) - 包括河流的长度，以及流经过的州
 - 山 (mountain) - 山的高度，以及山所在的州
-

下面是该小型数据库所对应的几个实例：

```
state('alabama','al','montgomery',3894.0e+3,51.7e+3,22,'birmingham','mobile',
      montgomery,huntsville)
city('california','ca','san francisco',678974)
river('colorado',2333,['colorado','utah','arizona','nevada','california'])
mountain('alaska','ak','hunter',4442)
```

在上述小型知识库的基础上，GEO 数据集包括 880 个查询样例，每个样例使用逻辑表达式进行标注，逻辑表达式的形式先后有 FunQL¹²、Prolog 和 lambda-表达式¹³。下面展示了该数据集中的一个问句，以及使用不同逻辑表达式的形式进行语义表示的例子。

句子: *What is the highest point in Florida?*

```
FunQL:answer(highest(place(loc(stateid('florida')))))
Prolog:answer(A,(highest(A,(place(A),loc(A,B),const(B,stateid(florida))))))
lambda-表达式: argmax( x.place(x) loc(x,florida), x.elevation)
```

GEO 是语义解析领域广泛使用的数据集。甚至还被翻译成其它语言 [85, 130]，也有中文翻译版本，可以用于测试中文语义解析 [161, 162]。该数据集总共有 280 个不同的词语，句子的平均长度是 8.5。880 个样例中 600 个样例作为训练数据，280 个样例作为测试数据。评价标准是：如果系统给出的逻辑表达式与给定的逻辑表达式完全匹配则为正确，否则为错误。该数据集主要有如下特点：

⁹简称 GEO，在本文的后续章节，我们都使用 GEO。

¹⁰<http://www.cs.utexas.edu/users/ml/geo.html>

¹¹<https://en.wikipedia.org/wiki/Prolog>

¹²<http://www.cs.utexas.edu/ml/wasp/geo-funql.html>

¹³https://en.wikipedia.org/wiki/Lambda_calculus

- 数据集的知识库本体比较简单，本体中谓词（对应到数据库中的关系）的数量不是太多。
- 问句的结构相对来说比较复杂，大部分问句都存在几个关系的串联，并不是仅仅只包含一个关系和一个实体。
- 问句中的语言现象比较丰富，包括串联（and 或者 or）最高级、比较级、否定等；句型也比较多样，存在从句、介词短语提前等。

2.3.5.2 JOBS 数据集

JOBS 数据集最早出现在 Tang and Mooney [132]，该数据集是跟工作相关信息查询的数据集。该数据集总共有 388 个词语，句子的平均长度是 9.8。数据集包括 640 个查询样例，每个样例使用逻辑表达式进行标注，逻辑表达式的形式有 Prolog 和 lambda-表达式。640 个样例中 500 个样例作为训练数据，140 个样例作为测试数据。评价标准与 GEO 数据集一样。下面展示了该数据集中的一一个问句，以及使用不同的逻辑表达式进行语义表示的例子：

句子: *Are there any jobs using Java that are not with IBM?*

Prolog: `answer(A, (job(A), language(A, java), \company(A, ibm)))`

lambda-表达式: `x.job(x) language(x, java) \company(x, ibm)`

2.3.5.3 ATIS 数据集

ATIS 数据集最早在 Dahl et al. [38] 中提出。He and Young [60] 在语义解析任务中正式使用该数据集。该数据集是跟美国航空相关查询的数据集。数据集包括 5418 个查询样例，每个样例使用 lambda-表达式作为逻辑表达式进行标注。下面展示了该数据集中的一一个例子：

句子: *what flights are there from minneapolis to newark on continental*

lambda-表达式: `(lambda $0 e (and (flight $0) (airline $0 co:al) (from $0 minneapolis:ci) (to $0 newark:ci)))`

该数据集主要有如下特点：

- 数据比较庞大，但是数据背后的知识库本体比较简单，谓词的总数量也不多。
- 句子中实体数目比较多，一个句子中可能出现多于 3 个实体。
- 句型的变化比较少，但是句子的长度比较大，一个句子中的谓词数量比较多，有不少是多于 3 个。

2.3.5.4 FREE917 数据集

FREE917 数据集最早在 Cai and Yates [21]中公布。该数据集以 Freebase 为知识库本体，与前面的数据集不同的是该数据集覆盖到多个领域。数据集包括 917 个查询样例，通常，917 个样例中 641 个样例用于训练数据，276 个样例用于测试数据。每个样例都使用 lambda-表达式作为逻辑表达式进行了标注，后来 Berant et al. [16]使用 lambda-DCS 对该数据集重新进行标注。下面给出的是该数据集中的示例：

句子:*what major events in history happened in africa*
 lambda-表达式:(lambda \$0 /common/topic (/location/location@events:t
 /en/africa:/location/location \$0))
 lambda-DCS: (!fb:location.location.events fb:en.africa)

该数据集主要有如下特点：

- 数据覆盖多个领域，数据以 Freebase 为知识库本体，本体中谓词的数目非常多。
- 句子的结构不是很复杂，大部分只包含一个实体和一个关系词，也有包含描述事件的。

2.3.5.5 WEBQUESTIONS 数据集

WEBQUESTIONS 数据集最早在 Berant et al. [16]中公布。该数据集以 Freebase 为知识库本体，同样覆盖多个领域。数据集包括 5800 个数据样例，5800 个样例中，3778 个样例用于训练数据，2023 个样例用于测试数据，每个样例都标注了答案。下面给出的是该数据集中的示例：

句子:*what is the name of justin bieber brother?*
 答案: (list (description "Jazmyn Bieber") (description "Jaxon Bieber"))

WEBQUESTIONS 数据集是当前研究工作中最常用的数据集，主要因为它是目前可用于语义解析唯一的大规模、且面向开放域的数据集。该数据集主要有如下特点：

- 数据覆盖多个领域，数据以 Freebase 为知识库本体，本体中谓词的数目非常多。
- 句子的结构不是很复杂，大部分只包含一个实体和一个关系词，也有包含描述事件的。

- 标注形式是答案，没有标注好的逻辑表达式。
- 数据规模较大。

2.3.5.6 OVERNIGHT 数据集

OVERNIGHT 数据集最早在 Wang et al. [139]中公布。该数据集包含 8 个领域，每个领域都有它单独的知识库本体，且每个知识库本体中谓词的数目比较少。该数据集中的样例都使用基于 lambda-DCS 的逻辑表达式进行标注。数据集中的 8 个领域以及各个领域中的数据情况统计如表2.2所示：

Domain	# example	# train	# test
Social Network	4419	3535	884
Blocks	1995	1596	399
Basketball	1952	1561	391
Restaurants	1657	1325	332
Calendar	837	669	168
Housing	941	752	189
Publications	801	640	161
Recipes	1080	864	216

表 2.2: OVERNIGHT 数据集中各个领域的数据统计

2.3.5.7 IFTTT 数据集

IFTTT 数据集最早在 Quirk et al. [121]中公布。该数据集都是基于“如果（条件），那么（动作）”的形式，数据都是从 IFTTT 网站¹⁴抽取而来。数据中每个样例都有两部分标注，一部分是条件的标注，即触发条件；另一部分是动作的标注。数据集中总共有 552 个触发函数，以及 229 个动作函数。该数据集中 77495 个样例作为训练数据，5171 个样例作为开发数据，4294 个数据用于测试数据。该数据集主要有如下特点：

- 数据规模非常大，虽然都是基于 If-This-Then-That 的句式，但是覆盖到的领域多达 128 个。
- 与以往的基于知识库的查询数据不同，该数据集主要是和操作指令相关的。

¹⁴<https://ifttt.com/>

2.3.5.8 WIKITABLEQUESTIONS 数据集

WIKITABLEQUESTIONS 数据集¹⁵最早在 Pasupat and Liang [115]中公布。该数据集以 Wikipedia¹⁶网站中的表格为知识库本体。每一表格有若干问题，每个问题的标注了从表格中获取的答案。该数据集共有 2108 个表格，22033 个问题，每个表格平均有 10.45 个问题。一般使用其中 20% 的表格及其对应的问题作为测试数据，其他的作为训练数据。测试数据中的表格都没有在训练数据中出现过。该数据集的特点是：

- 数据规模非常大，以表格为知识库本体。知识库本体中总的谓词数量较大，共有 3929 个。
- 问句的结构非常复杂，基本都是复合型的句子，每个问句中包含的谓词数量基本都超过 2 个。

2.4 语义解析方法分类

现阶段大部分语义解析所考虑的解析对象都是一个句子，这种情境，句子是独立的单元，不需要其他的上下文。还有另外一种情境，句子之间相互有联系，如基于交互式的查询情境，用户输入一个问题，计算机理解这个问题，并给出一个答复，用户在上一个问题的基础上再添加一些条件，此时计算机理解这个句子就要考虑上文信息，一般是这个句子里面有指代词，如“ones”，图2.13展示了这样一个例子，该例子是 ATIS 数据集中的一个例子。针对这种情境的语义解析，代表工作有：Zettlemoyer and Collins [153]利用 CCG，并设计特殊的文法和机制来处理指代的问题。Iyyer et al. [67]针对类似的问题，在 Yih et al. [149]所提出的 STAGG 方法的基础上构建了一个动态的基于神经网络模型的语义解析框架，具体细节在这里不细述。

- (a) show me the flights from boston to philly

$$\lambda x. flight(x) \wedge from(x, bos) \wedge to(x, phi)$$
- (b) show me the ones that leave in the morning

$$\begin{aligned} \lambda x. & flight(x) \wedge from(x, bos) \wedge to(x, phi) \\ & \wedge during(x, morning) \end{aligned}$$
- (c) what kind of plane is used on these flights

$$\begin{aligned} \lambda y. \exists x. & flight(x) \wedge from(x, bos) \wedge to(x, phi) \\ & \wedge during(x, morning) \wedge aircraft(x) = y \end{aligned}$$

图 2.13: ATIS 数据集中有上下文的并用 lambda-表达式进行语义表示的例子

¹⁵<https://nlp.stanford.edu/software/sempre/wikitble/>

¹⁶<https://www.wikipedia.org/>

带上下文的语义解析除了上述情境外，还有基于多句子或者篇章的语义解析，Lee et al. [88]提出了利用语义解析方法对篇章里面的时间短语进行解析的，大部分情况都需要利用上下文才能对篇章中的时间短语进行准确的解析，而他们所提出的语义解析方法也是能够利用上下文信息的。

在本文中，我们考虑的解析对象是一个句子。接下来对语义解析方法进行分类，这些方法的解析对象也是一个句子。最早的语义解析系统基本都是基于规则的，随着机器学习算法的兴起，基于机器学习的语义解析 [73, 55] 逐步替代了基于规则的语义解析，基于机器学习的语义解析方法不再需要定义规则，而是通过从训练语料中学习规律。近年来，深度学习模型在图像领域取得了成功，研究人员也开始在自然语言处理的各种任务中尝试使用深度学习模型，语义解析任务也不例外。基于深度学习的语义解析方法是端到端的，可以不需要定义词汇特征、文法特征等。仔细分析上述语义解析的各种方法，在本文中我们将其分为 5 类：基于规则的语义解析、基于组合文法的语义解析、基于机器翻译的语义解析、基于语义图的语义解析和基于序列预测的语义解析。下面我们将逐一对每一类方法进行说明，并比较各类方法的优点和不足。

2.4.1 基于规则的语义解析

在机器学习方法复兴之前，大部分自然语言处理的系统都是基于规则的。语义解析任务中，有两个著名的基于规则的系统。它们都是在特定的情境下，利用特定的规则对自然语言句子进行解析。一个是 Bobrow [18] 设计的 STUDENT 系统，这是一个利用规则解决算术问题的系统，下面是该系统中的几个例子：

Example: *The sum of two numbers is 111. One of the numbers is consecutive to the other number. Find the two numbers.*

Example: *Bill's father's uncle is twice as old as Bill's father. 2 years from now Bill's father will be 3 times as old as Bill. The sum of their ages is 92. Find Bill's age.*

Example: *The distance between New York to Los Angeles is 3000 miles. If the average speed of a jet plane is 600 miles per hour find the time it takes to travel from New York to Los Angeles by jet.*

另一个是 Woods et al. [142] 提出的 LUNAR，一个关于月球岩石的问答系统，下面是该系统中的几个问题：

Example: *What is the average concentration of aluminium in high alkali rocks?*

Example: *What samples contain P205?*

Example: *Give me the modal analyses of P205 in those samples.*

这类系统的特点是：即使是特定领域，规则的规模也很庞大，需要消耗很大的人力物力来构造覆盖该领域中可能出现的语言现象，模型鲁棒性也很差，十分脆

弱，也无法从一个领域迁移到另一个领域。机器学习兴起后，基于规则的系统就迅速被基于机器学习的系统所取代。

2.4.2 基于组合文法的语义解析

基于组合文法的语义解析可以说是最符合组合语义学和模型理论语义学的基本准则的，这类方法一般使用词典来表示句子中各个成分的意思，实现语义落地，再利用组合规则来组合各个成分的意思。在发展的过程中，有三种组合文法产生了广泛的影响，它们分别是：同步文法（SCFG），组合范畴文法（CCG）和基于依存的组合语义文法（DCS），另外还有一种基于混合树的方法。接下来，我们将分别对这几种文法进行介绍。

2.4.2.1 同步文法（SCFG）

同步文法（synchronous context-free grammar, SCFG）最早是用于编译程序语言 [1]，后来被用于机器翻译中 [143]，该文法特别适合基于结构的机器翻译，因为它不仅能够包含源句子和目标句子的句法结构，还能够建模源句子和目标句子中每一个子块的映射关系，该文法的有效性已经在机器翻译中得以验证 [27]。语义解析与机器翻译类似，都需要把一种语言翻译成另外一种语言。Wong and Mooney [140]最先将该文法用于语义解析中。类似于机器翻译，语义解析中使用同步文法，可以同时利用句子和语义表示的结构，最终同时得到句子和句子的语义表示，有充分考虑句子中词语和语义表示中的谓词的映射关系。后续还有工作对同步文法进行扩展来增强语义解析 [141, 89]。

2.4.2.2 组合范畴文法（CCG）

组合范畴文法（CCG）最早用于句法分析 [30, 63]，由于 CCG 中的文法都是从语言本身抽象出来的，能够覆盖很多的语言现象，这在句法分析中已经得以验证 [29]。另外，CCG 文法中的范畴（category）虽然只包括句法信息，但是与语义的耦合性非常好，前面已经介绍过，及物动词“border”的句法形式是： $(S \setminus NP) / NP$ ，这表示该动词还需要搭配一个宾语（右侧）和一个主语（左侧），而它的语义表示为：

. $y \cdot \text{border}(y, x)$ ，其中 y 表示宾语的语义， x 表示主语的语义。这种将 CCG 句法结构为主体，再加上语义信息的方式最早由 Zettlemoyer and Collins [154]提出。CCG 主要有词典和组合规则组成，词典定义了词语的句法表示和语义表示，文法主要使用几条常用的组合文法（前面已介绍），使用 CCG 进行语义解析的方法的核心是学习 CCG 中的词典，后续有很多工作都集中于使用不同算法学习覆盖度更广、质量更高的词典。也有工作为了使语义解析能够用于特定的语言现象，如口头

类的文字中容易出现缺省、词序不规范等现象，提出在原有组合规则的基础上添加几条新规则，如 Zettlemoyer and Collins [152]。

使用 CCG 进行语义解析的优势是：CCG 十分契合组合语义的基本准则，同时 CCG 本身的表示能力强，能够覆盖大部分语言现象，还能够根据特定的语言现象，对组合文法进行调整。缺点是：表示过于复杂，句子中的每一个词都具有相应的句法和语义表示，特别是句子中只在句法中起连接作用，而没有一点语义的词语，其表示非常复杂，这些词本来在语义解析中是可忽略不管的，但是由于 CCG 中句法和语义是同步的，导致不得不为每一个词都赋予句法和语义表示，这也使得解析过程中搜索空间很大。

2.4.2.3 基于依存的组合语义文法（DCS）

基于依存的组合语义文法（dependency-based compositional semantics, DCS）最早由 Liang et al. [93]提出，相比 CCG，DCS 的优势是：DCS 的表示是树结构的，类似于依存树，有利于借用依存分析中的方法来解析和学习；DCS 的形式非常简单，同时也有很强的表示能力；另外 DCS 的表示非常有利于执行（execute），特别是存在量词的情况 [94]。为了进一步增强 DCS 的语义表示能力，Liang [91]提出了 λ -DCS，现在我们说 DCS，一般是指 λ -DCS。

DCS 中的项要么是一元的（*unary*, 用 u 表示），常表示一个集合，要么是二元的（*binary*, 用 b 表示），常见的一元形式就是实体，如 `Seattle`，常见的二元形式就是二元谓词（*predicate*），如 `PlaceOfBirth`。DCS 中还有几条组合规则，如结合（join）和相交（intersection），结合规则用于组合一元形式和二元形式，如组合 `Seattle` 和 `PlaceOfBirth` 得到 `PlaceOfBirth.Seattle`，得到的项也是一个一元的项。相交规则用于组合两个一元形式，如 `Profession.Scientist` \sqcap `PlaceOfBirth.Seattle`。当然还有一些规则用于建模语言中的其他现象，如 count 规则，作用于一元项，用来统计一元项所表示的集合中元素的个数。

2.4.2.4 混合树（Hybrid Trees）

Lu et al. [98]最早提出一种基于混合树（Hybrid Trees）的语义解析方法，他们将语义解析过程看成是混合树的生成过程，并用一个生成模型来建模这个过程。有点类似同步文法，混合树的节点既包含句子中的词语，也包含语义单元。不同的是，使用同步文法的方法 [140] 利用自底向上的方法进行解析，这意味着需要学习词汇。而使用混合树的方法的解析过程是自顶向下的，解析的具体实现借鉴了 Collins [34]在句法分析中的使用的算法。后续还有工作对混合树进行改进 [96, 97]。

2.4.3 基于翻译的语义解析

机器翻译是用计算机将一种自然语言（源语言）自动翻译成另一种自然语言（目标语言），语义解析是把自然语言句子（源语言）自动解析成计算机可识别的语义表示（目标端），它们都是一个翻译（或者映射）的过程，源端的语言和目标端的语言都具有结构性，两者之间都存在对齐（alignment）。有研究者直接把语义解析看作是机器翻译问题，利用机器翻译的方法来进行语义解析 [140, 2]。

2.4.4 基于语义图的语义解析

前面已经提到语义表示的方法可以使用框架来进行表示，这种表示方法的优势是利用了自然语言句子的句法结构与语义表示的结构之间存在的相似性。基于语义图的语义解析方法主要分三步：第一步构建结构图（称为语义图），第二步进行语义落地，即将图中的节点和边赋予语义信息；第三步将已经完成语义落地的语义图转换为逻辑表达式。

相比基于组合文法的语义解析方法，基于语义图的语义解析方法的优势是：不需要词典，也不需要组合文法。基于语义图的语义解析方法的关键在于如何有效的构建语义图。现阶段语义图的构建有如下几种：从句法树构建语义图、从依存树构建语义图、从模板匹配构建语义图、和利用启发式规则构建语义图。下面我们将分别介绍这几种语义图构建方法。

2.4.4.1 从句法树构建语义图

句法分析的发展已有很长的历史，在句法分析的基础上进行语义解析是顺理成章的尝试，但也存在一定的挑战，即如何把叶子节点的语义组合成最终的句子语义表示，Ge and Mooney [54]利用词典来处理叶子节点的语义，利用组合规则来组合较小孩子节点的语义，由于句子的句法树结构与语义结构容易出现异构的情况，因此他们提出了超谓词（macro-predicate）的概念来处理这种异构。类似的工作还有 Ge and Mooney [53]和 Krishnamurthy and Mitchell [82]

2.4.4.2 从依存树构建语义图

相比句法树，依存树更侧重于句子中各成分的语义关系，也更适合用来进一步转换成表示语义的语义图。Zou et al. [158]利用句子的依存树来构建查询图。Reddy et al. [124, 125]把句子的依存树转换为句子的语义图。

2.4.4.3 从模板匹配构建语义图

以模板匹配的方法来构建语义图的方式只适合在特定的情境下使用。如，句子比较简单，句子的结构比较固定，且变化比较少，这样可以定义若干个模板来覆盖到大部分句子的语义结构。Bast and Haussmann [12]针对基于 Freebase 的事实类问题，设计了三个模板来覆盖大部分的问题形式。图2.14展示了模板以及相应的例子。

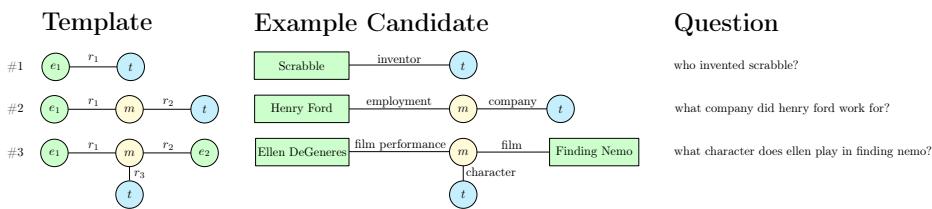


图 2.14: 查询模板及利用模板进行语义表示的例子

图2.14所示例子的模板中， e 表示实体， r 表示关系， t 表示结果实体， m 表示一个中间值，这是 Freebase 中特有的，没有具体的意义。基于这种方法的优势是搜索空间很小，目标语义图的结构总共才三种，很好进行匹配。而缺点也很明显，就是推广性很差，很难处理新的情境和复杂的情境。

2.4.4.4 利用启发式规则构建语义图

这种方法是一种受限制的启发式的逐步构建语义图的方法，Yih et al. [149]最早提出这种方法。该方法总共分为三步：第一步实体链接，第二步关系匹配，第三步加入约束条件。整个过程语义和句法结构同步进行，句法结构直接基于目标知识库本体结构而构建。这种方法的优势是，不必担心句法结构与语义结构异构，而缺点就是搜索空间取决于目标知识库的大小，像以 Freebase 为知识库本体的情况，搜索空间一般很大。

2.4.5 基于序列预测的语义解析

随着深度学习在自然语言领域获得广泛运用，越来越多的研究者开始尝试使用深度学习模型来建模自然语言处理中的问题。比如运用卷积神经网络模型在分类问题上，运用循环神经网络模型在序列预测任务上。把逻辑表达式序列化，语义解析就可以看作是一个序列预测问题，进而尝试使用循环神经网络模型来进行语义解析 [144, 43, 68, 13, 26, 61]。这种方法的优势是不需要词典和文法，也不需要依赖于依存分析、特定模板或启发式规则，也不需要设计特征，整个过程是端到端的，方便模型的学习。这种方法的缺点是把所有的结构预测和语义落地都交给神经网络模型来完成，这个过程可能会遗漏一些信息，特别是结构的信息。

2.5 近期语义解析的应用举例

这一节，我们将介绍近期几种典型的语义解析的应用研究。

2.5.1 语言指令控制的机器人

机器人控制指的是人类用自然语言给机器人下指令，机器人执行相应操作。这需要机器人能够理解人类的语言，把自然语言指令解析成机器可识别的操作指令。具体的，Artzi and Zettlemoyer [4]设置了这样一个情境，把机器人放在一个空间里，该空间里面有其他的物体，如墙壁，沙发等。在这样的情境下，人给机器人发布一些自然语言的指令，机器人对这些指令进行解析后，执行相应的操作。图2.15展示了利用 lambda-表达式来表示指令的语义的例子。类似场景下的研究工作还有 Matuszek et al. [100], Chen and Mooney [25]。

```

move forward twice to the chair
 $\lambda a. move(a) \wedge dir(a, forward) \wedge len(a, 2) \wedge$ 
 $to(a, \iota x. chair(x))$ 
at the corner turn left to face the blue hall
 $\lambda a. pre(a, \iota x. corner(x)) \wedge turn(a) \wedge dir(a, left) \wedge$ 
 $post(a, front(you, \iota x. blue(x) \wedge hall(x)))$ 
move to the chair in the third intersection
 $\lambda a. move(a) \wedge to(a, \iota x. sofa(x)) \wedge$ 
 $intersect(order(\lambda y. junction(y), frontdist, 3), x)$ 

```

图 2.15: 使用 lambda-表达式标注指令的语义的例子

2.5.2 解代数应用题的机器人

Kushman et al. [84]提出利用语义解析技术来解代数应用题，图2.16展示了一个例子。同样 Hopkins et al. [64]也利用语义解析技术来求解数学 SAT 考试中的代数题。

Word problem	An amusement park sells 2 kinds of tickets. Tickets for children cost \$ 1.50 . Adult tickets cost \$ 4 . On a certain day, 278 people entered the park. On that same day the admission fees collected totaled \$ 792 . How many children were admitted? How many adults were admitted?	
Aligned template	$u_1^1 + u_2^1 - n_1 = 0$	$n_2 \times u_1^2 + n_3 \times u_2^2 - n_4 = 0$
Instantiated equations	$x + y - 278 = 0$	$1.5x + 4y - 792 = 0$
Answer	$x = 128$	$y = 150$

图 2.16: 利用语义解析解代数应用题的例子

2.5.3 计算机操作语言助手

语义解析能够促进人机交互的实现。具体的，人类直接输入自然语言的指令，而不是标准的形式化的机器操作指令（如 bash 指令），并且机器能够理解自然语言的指令，并进行相应的操作。Lin et al. [95]利用语义解析技术构建 Linux 操作系统的交互式接口，用户可以输入自然语言的指令，接口背后的语义解析器可以将自然语言指令转换成形式化的指令，图2.17展示了自然语言指令和对应的形式化的机器指令。

Natural Language	Bash Command(s)
<i>find java files in the current directory tree that contain the pattern 'TODO' and print their names</i>	grep -l "TODO" *.java find . -name "*.java" -exec grep -il "TODO" {} \; find . -name "*.java" xargs -I {} grep -l "TODO" {}

图 2.17: 自然语言指令和对应的 bash 指令

2.6 本章小结

本章，我们首先介绍了语义解析的一些背景知识，以及研究语义解析所要了解的一些概念，包括语义表示形式，解析算法，学习算法，公开数据集。接着对当前语义解析方法进行分类：基于规则的语义解析、基于组合文法的语义解析、基于机器翻译的语义解析、基于语义图的语义解析和基于序列预测的语义解析，并介绍了每类方法中的代表性工作，还给出了每类方法的优缺点。最后介绍了近期语义解析的应用举例，如语言指令控制的机器人、解代数应用题的机器人和计算机操作语言助手。

第三章 基于词典扩展学习的语义解析

3.1 本章引论

基于组合语义的语义解析器一般包含两个核心组件：词典和文法。词典是词汇的集合，每一个词汇表示一个自然语言中的词语到知识本体中谓词（predicate）的映射，如 *born* :: Place_of_birth。通常，词汇分为三类：实体词汇、类别词汇和关系词汇。实体词汇对应到知识本体中的实体；类别词汇对应到知识本体中的一元谓词；关系词汇对应到知识本体中的二元谓词。这三类词汇的实例分别对应到图3.1中的词汇 2,、词汇 1 和词汇 3。基于词典中的词汇，语义解析器基于组合原理：“句子的意思是句子中各个成分的意思的组合”，使用组合文法把句子中的小的语义单元组合成最终的语义表达式。

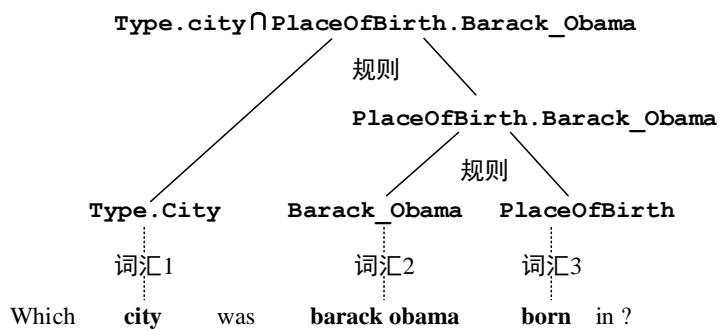


图 3.1: 基于组合语义的语义解析的例子

词典是基于组合语义的语义解析方法的基础。图3.1展示了一个语义解析的例子，从例子中我们可以看到，词汇作用于解析过程中的最底层，可以说没有词汇，就无法自底向上的解析。目前大部分的词典都通过词典学习的方式得到。最初的词典学习需要利用形式为（句子，逻辑表达式）对的训练语料，这里词语到谓词的映射类似于机器翻译里的对齐，可以使用类似 EM 算法的形式学习到词汇。这类方法的缺点是需要大量的标注语料，容易出现词典覆盖度不够的问题。

近年来，越来越多的语义解析从受限领域扩展到开放领域，进一步加剧了词典学习的挑战。传统的语义解析都是领域限定的，限定领域的知识库本体中谓词的数量很有限 [154, 85, 5]，在这种情况下，完全可以通过标注语料学习到词语到谓词的映射关系。但是随着语义解析技术逐渐面向开放域 [22, 21, 16, 79, 87]，目标知识库本体中的谓词（predicate）数目也越来越多，而自然语言本身表达具有丰富性：每一个意思可以有多个不同的词语来表达，这使得词典学习越来越困难，其中词典中词汇覆盖度问题尤其显著。如何加强词典学习，提高词典中词汇覆盖度成了语义解析领域的一大研究难点和重点。

词典学习¹一方面是语义解析任务中不可或缺的一部分，另一方面它本身也可作为自然语言处理中一项十分重要的任务，词典可用于机器翻译，信息抽取等。近年来，语义解析领域开始有很多工作着手于词典学习，Cai and Yates [22]提出了基于模板的词典学习方法，该方法利用文本语料和知识库中实体对的共现，以及特定模板的约束来学习词语到谓词之间的映射。具体地，他们基于知识库定义知识模板（knowledge pattern），从搜索引擎的结果中抽取文本模板（text pattern）。他们的方法基于如下假设：句子中两个实体之间的词语是这两个实体之间的关系，如果这两个实体存在于知识库中某个谓词的实例中，那么该词语应该映射到这个谓词。这种方法使用远距离（distant）监督学习算法能够学习到新的词汇来扩充词典，一定程度上提高了词典的覆盖度。但由于谓词数目实在太多，有很多谓词都无法依靠这种方法得到准确的词语与之匹配。Berant et al. [16]等在此基础上提出桥连接（bridge）的方法，该方法在没有词语触发二元谓词的情况下，也就是词典中的词汇没有覆盖到的情况，能够利用当前的解析结果信息（如实体已识别）和知识库本体的约束，自动引入二元谓词。该方法虽然没有学习新的词汇，但是桥连接充当了词汇的功能，在一定程度上能够舒解词典覆盖度不够的问题。然而，桥连接方法并没有学习到真正的词汇。

本章我们围绕词典的学习问题，展开词典扩充学习的研究。一方面现有方法还存在继续改进的地方，另一方面，这些方法仅仅利用了文本语料和知识库语料中实体对的共现来学习词典，由于语料的不完备，词典的覆盖度还是会有所限制，实际上，我们还可以利用文本与文本之间的联系来辅助扩充词典。本章的工作核心是利用词典扩充的方法来解决语义解析中词典覆盖度不够的问题。我们从两个不同方面开展工作，一个是对现有方法的改进，另一个是提出一种新的半监督的词典学习方法。

首先我们在现有工作的基础上提出了一种基于桥连接的词典学习方法。该方法在桥连接方法的基础上，在训练的过程中，每当自动引入一个二元谓词，我们就从句子中抽取非实体性的内容词语（疑问词、形容词、名词和动词），并与该二元谓词配对，作为新的词汇加入到模型中。随着新词汇的大量加入，词典中会存在很多带“歧义”的词汇，同时也存在很多无用的甚至错误的词汇。为了让模型选择更好的词汇，我们设计了新的词语-二元谓词特征模板，该特征模板能够更好的捕捉词语和二元谓词之间的映射关系。为了截枝无用的、错误的词汇，我们设计了基于投票机制的核心词汇获取方法，核心词汇虽然牺牲了一点语义解析系统的召回率，但是能够大大提升语义解析的效率，还能提高语义解析的准确率。

这部分工作的贡献总结如下：

¹ 在本文中，我们有时会说词典学习，有时也会说词汇学习（或者学习词典和学习词汇），词典偏集合，词汇偏个例，只有在特指个别映射的时候才能只说词汇，其他情况，词汇和词典通用。

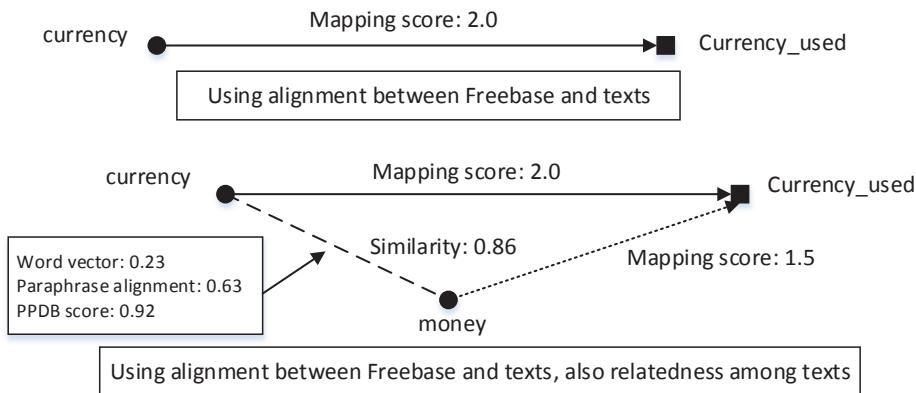


图 3.2: 对比我们的方法和之前的方法

1. 针对语义解析任务中词典覆盖度不够的问题，我们在现有工作的基础上提出了基于桥连接的词典学习方法。
2. 我们设计了新的词语-二元谓词的模板特征，来消除新词典中容易出现歧义的问题。
3. 我们设计了基于投票机制的核心词典获取方法来舒解新词典中存在无用或者错误词汇所带来的影响。

另外，我们还提出了一种新的基于半监督的词典学习方法。该方法利用容易获取的文本语料和词汇资源²来扩充词典。这种方法的出发点是：具有相同或相近意思的词语应该映射到相同或相近的谓词，而词语之间的相似性可以用来把已知的映射传播到未知的映射。比如，我们假设已经有了种子（seed）映射：“*currency*” :: *currency*，而我们知道词语“*money*”与词语“*currency*”密切相关，因此我们可以推断“*money*”也应该映射到“*currency*”。为了达到这个目标，我们利用一个基于图的半监督学习框架，该框架不仅利用 Freebase 和文本之间的实体对共现来学习词汇，还利用文本与文本之间的语义相关性来从种子词汇中学习新的词汇。我们也是首例利用文本之间的相关性来学习高覆盖度词典的。图3.2对比了我们的方法和之前的方法。图中上部分是之前的方法，该方法仅利用 Freebase 和文本之间的实体对共现来学习词汇；图中下部分是我们的方法，该方法不仅利用 Freebase 和文本之间的对齐，还利用利用文本与文本之间的联系。

这部分工作的贡献总结如下：

1. 我们提出一种新的基于半监督词典学习框架，该框架能够学习到高覆盖度的词典。与之前方法不同，该框架能够充分利用容易获取的大量文本语料和词典资源。

²在本章中，我们用文本语料（text corpora）来指代网络上的文本资源，如 Wikipedia 和 WikiAnswers；用词典资源（lexical resources）来指代收集整理好的词典有关的资源，如 WordNet 和 PPDB。

2. 我们设计了一个基于图的半监督学习算法，该算法能够从少量的种子词汇中学习到该覆盖度的词典。

我们在两个公开数据集（WEBQUESTIONS 和 FREE917）上进行了实验，实验论证了我们两个不同方法的有效性：我们的方法都超过了基准系统，特别是大大提高了基准系统的召回率，这得益于扩充后的词典覆盖度更高。

3.2 基于桥连接的词典学习的语义解析

基于桥连接的词典学习是对现有工作的一个改进，该方法在桥连接的基础上学习新的词汇，扩充得到新的词典，同时利用策略提升词汇的准确率。新词典会运用于语义解析中。这部分的框架图如图3.3所示。接下来，我们将介绍我们使用的基准语义解析器，并介绍该解析器所运用的桥连接方法和针对桥连接的词语-谓词特征模板，然后介绍我们在这个基础上提出了基于桥连接的词典学习的语义解析方法。具体包括基于桥连接的词典学习方法，和两个用于提升词汇准确率的策略：通用的词语-谓词特征模板和基于投票机制的核心词典获取。

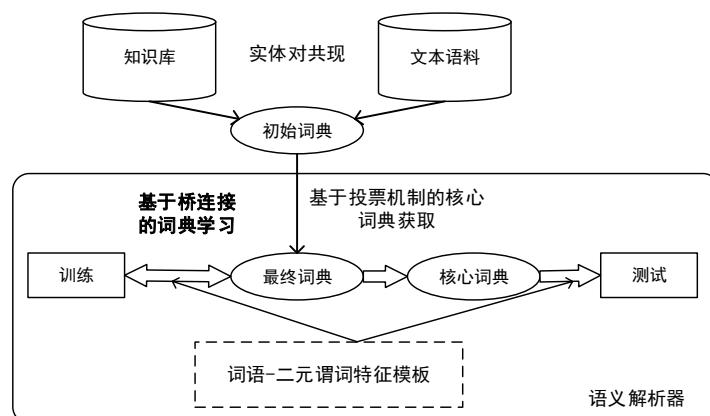


图 3.3: 基于桥连接的词典学习的语义解析方法框架图

3.2.1 基准语义解析器

这一节介绍我们用于对比系统的基准系统，我们采用基于传统组合语义的语义解析系统作为我们的基准系统。具体的，我们的系统都基于 SEMPLRE 系统³而搭建。该系统总共由 5 个模块 [92] 组成：文法、模型、解析器、学习器和执行器（如图3.4所示）。其中文法是该系统的基础，也是核心，该模块包括词典和组合文法；解析器负责利用文法对句子进行解析；模型模块负责对解析器所得到的候选解析结果进行打分排序；学习模块负责利用训练数据对模型模块中的模型参数进行

³<https://nlp.stanford.edu/software/sempre/>

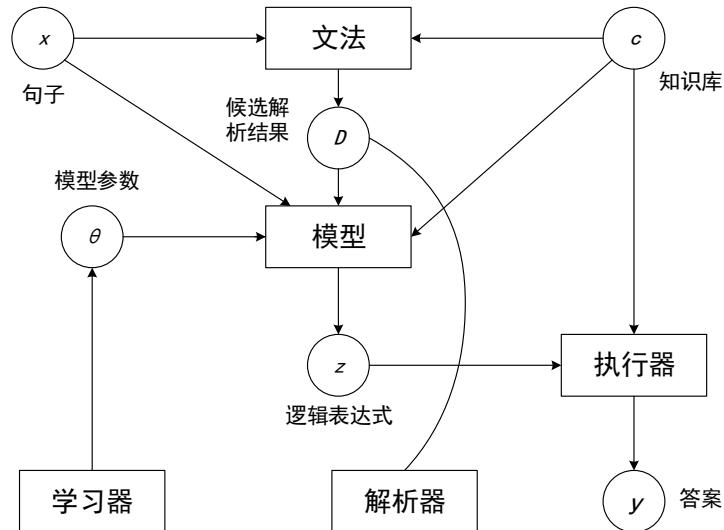


图 3.4: 基于组合语义的语义解析方法的基本框架图

学习；执行模块负责利用知识库对得到的逻辑表达式进行求值，得到句子所对应的答案。

3.2.1.1 桥连接

Berant et al. [16]提出了桥连接方法来充当词汇的功能。具体的，当句子中的词语没有触发任何谓词的时候，该方法可以根据当前的解析结果自动引入二元谓词，如图3.5例子所示，二元谓词 `currency`（通用货币）并不是由关系词汇触发而来，而是利用当前所得到的实体 `SriLanka`（斯里兰卡，一个国家）自动引入的（所有与这个实体关联的二元谓词都会引入，模型再利用特征来选择最合适）。该方法也是用来解决词汇覆盖度不够的问题。但是它并没有学习到真正的新词汇。

3.2.1.2 针对桥连接的词语-谓词特征模板

Berant and Liang [15]针对桥连接设计了一个词语-二元谓词的特征模板 (`lemmaAndBridge`)，该特征模板在使用桥连接的时候激活，该特征模板用来描述句子中非实体性内容词（疑问词、动词、形容词和名词）与桥连接所引入的二元谓词之间的映射关系。如图3.5例子所示，引入的二元谓词是 `currency`，句子中的内容词是“`what`”、“`money`”和“`take`”，这样就会触发三个特征：“`what` —— `currency`”、“`money` —— `currency`”和“`take` —— `currency`”。桥连接与该特征模板的联合使用可以说是充当了隐式词汇的功能。

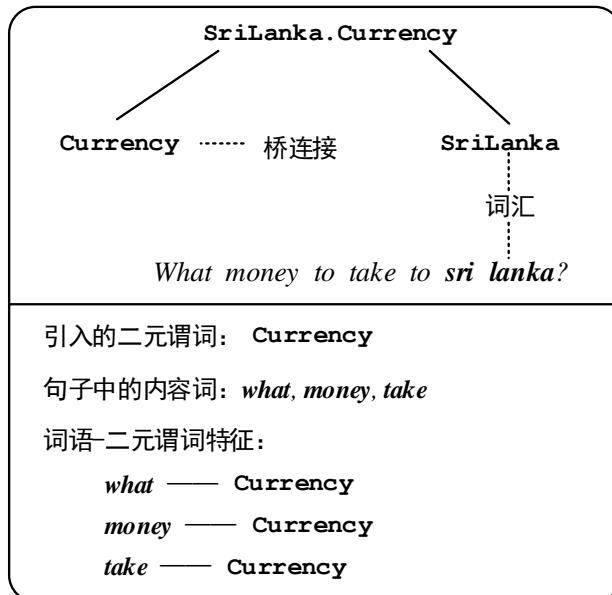


图 3.5: 桥连接方法示例

3.2.2 基于桥连接的词典学习的语义解析

这一节介绍我们提出的基于桥连接的词典学习的语义解析方法，首先我们介绍基于桥连接的词典学习方法，再介绍我们所采用的用于提高词典准确度的两个策略：新的词语-二元谓词的特征模板和基于投票机制的核心词汇获取。

3.2.2.1 基于桥连接的词典学习方法

为了解决词汇覆盖度不够的问题，我们在现有工作的基础上，提出基于桥连接的词典学习方法。Berant et al. [16]提出的桥连接方法，能自动引入二元谓词，但他们的方法并没有学习到新的词汇，只是利用桥连接来充当词汇的角色。我们的方法在他们方法的基础上加以改进，可以引入新的词汇，并加以学习。具体的，在解析的过程中，当桥连接被激活的时候，会有一个二元谓词自动引入，而这个二元谓词并不是由句子中的某个词语触发的。而实际上，这个二元谓词本应该由句子中的某个词语触发，也就意味着应该存在一个类别词汇，这个词汇是句子中某个词语到这个二元谓词的映射（图3.5中的例子，本应该存在“*money* :: *currency* 这个词汇），而正因为词典的覆盖度不够，词典缺失了很多词汇，使得在解析的过程中，不能够利用词汇触发来引入谓词，为此，我们在这个过程中引入新的词汇，并把新的词汇加入到当前的词典中。当一个桥连接激活的时候，我们从句子中抽取非实体性的内容词语（疑问词、动词、名词和形容词），把这些词语分别与引入的二元谓词配对，形成若干新的词汇，图3.5的例子中会加入 3 个新词汇，分别是：

这些词汇中肯定包含了本应该存在的能够触发该二元谓词的词汇。我们把这些词汇加到现有词汇中，通过一定的样例训练以后，新加入的词汇中那些有效的词汇就

```
what :: currency  
money :: currency  
take :: currency
```

会被模型通过特征挑选出来，并可用于后面的对其它例子的解析。

词典中加入了新的词汇，并且基于桥连接加入的词汇中只有一部分是准确的，而另外一部分都是不准确的，这部分不准确的词汇不仅影响语义解析系统的准确性，还影响其效率。为此，我们设计了新的词语-二元谓词的特征模板，还设计了基于投票机制的核心词汇获取方法，用来提高词典的准确率和解析系统的效率。

3.2.2.2 通用的词语-谓词特征模板

Berant and Liang [15]设计的词语-二元谓词的特征模板仅在通过桥连接引入二元谓词的情况下激活，实际上，这个特征模板对所有的二元谓词都是十分有效的，不管这个二元谓词是通过关系词汇引入的，还是通过桥连接方法引入的。并且我们的方法通过桥连接方法学习到新词汇以后，我们在解析的时候将不再使用桥连接方法，具体的，第一轮训练的时候启用桥连接方法，用于引入新词汇，之后的训练和测试，都不再启用桥连接方法，这样就是想测试我们新学到的词汇的效果。为了让这个特征模板发挥更大的功效，我们设计了可用于所有二元谓词的词语-二元谓词的特征模板（lemmaAndBinary），该模板只要有二元谓词引入就会激活。

3.2.2.3 基于投票机制的核心词典获取

由于我们现在使用的词典很大，特别是采用了基于桥连接的词汇引入方法之后，并且词典中的很多词汇都是错误的，这给语义解析带来很大的困扰，不仅大大影响解析的效率，也影响了解析的性能。为此，我们提出了基于投票机制的核心词汇获取方法，该方法旨在获取当前词典中有用的、准确度高的核心词汇。Artzi et al. [5]也提出了类似的方法，用来压缩词典。

具体的，在训练的时候，对于每一个样例，模型利用当前的词典对此样例进行解析，在正确的解析中所使用到的类别词汇和关系词汇，我们对这些词汇做记一票处理，由于模型可以利用这些词汇得到准确的解析结果，我们认为这些词汇是重要的且准确的词汇。经过若干轮的训练，票数越多的词汇代表越重要越准确。为了抽取出核心词汇，又为了保障每一个谓词都存在它的核心词汇，首先我们从词典中抽取票数超过 1 的词汇，然后对这些词汇以谓词为基准进行分组并按照投票数进行排序，最后我们对每一个谓词都抽取投票数 top K 的词汇用来组成我们的核心词典。在这里我们没有使用概率的因素，而是直接利用硬性条件来进行刷选。我们认为一个词汇的投票数超过 1 就表明该词汇在某个例子的解析中充当了正确

的角色，应该纳入到核心词典的考虑范围；之所以为每一个谓词选择 top K（实验中 $K=5$ ）的词汇，是因为我们认为有 K ($K=5$) 个不同的词语来表达一个谓词的意思，就算自然语言表达具有丰富性的特点，也已经具有一定的覆盖度了。测试的时候，模型就使用核心词典来对测试样例进行解析。

3.3 基于半监督的词典学习的语义解析

这一节将介绍我们提出的基于半监督的词典学习的语义解析方法。该方法在一个基于图的半监督学习框架上，充分利用容易获取的文本语料和词典资源学习词语与词语之间的相似性，从少量种子词汇中学习新的词汇，再把扩充后的词典运用于语义解析器中。图3.6展示了我们这种方法的框架图。

3.3.1 基于半监督的词典学习

在面向开放域的情形下，很难从标准语料中学习到覆盖度足够的词典。在这节的工作中，我们利用若干（词语，谓词）的映射作为种子词汇，然后利用词语之间的形似度从已知映射的词语传播信息给旁边未知映射的词语，从而学习到新的（词语，谓词）的映射。具体的，我们提出一个基于图的半监督学习方法来实现这个目标。我们的方法能够利用大量容易获取的文本语料和词典资源来学习词语与词语之间的相似度，然后利用基于图的半监督学习算法在词汇图中进行传播平滑。这样一来，我们可以从少量种子词汇中，学习到新的词汇。

基于图的半监督学习方法广泛应用于自然语言处理的其他领域，在机器翻译领域主要被用来解决 OOV 问题 [122, 126, 157, 101]，还被应用于框架语义解析 [39]，情感词典学习 [57]，以及基于词型变化的词典学习 [49]。

我们利用基于图的半监督学习方法来扩充我们的词典，我们的方法主要分为如下三个步骤：

1. 种子词典构建：首先利用 Freebase 和文本语料之间的实体对共现来学习一个原始种子词典，并利用打分机制对该原始种子词典中的每一个词汇进行打分，

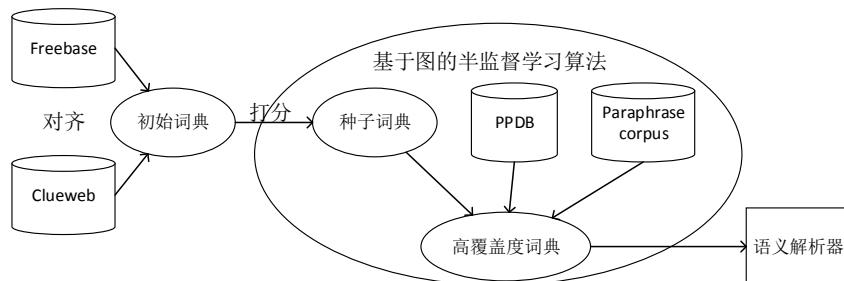


图 3.6: 基于半监督的词典学习的语义解析框架图

从中选择打分高的作为我们最终的种子词典。

2. 图构建：利用文本语料和词典资源学习词语之间的相似性，并构建用于图传播的图。
3. 图传播：利用种子词典和词语与词语之间的相似性在图传播算法的框架上进行新词典的学习。

接下来，我们将依次介绍这三个步骤。

3.3.1.1 种子词典构建

我们需要种子词汇来传播信息到新的未知的词汇。然而我们没有足够多的标注好的逻辑表达式来学习种子词典，另外，目标知识库本体中谓词的数量太多，我们也不能人工构建一个种子词典。为了构建高质量的种子词典（每个谓词都应该至少存在不少于一个种子词汇中），我们分两步来完成：

首先，我们利用 Freebase 和文本语料之间实体对的共现来构建一个粗糙的种子词典，方法与 Berant et al. [16]的类似，不同的是，在词语端我们只使用单词 (unigram)。这样做的原因有两个：一个是因为一个词语（超过一个词）能够触发某个谓词，那么由这个词语分出来的单词也能够触发这个谓词，从而不会牺牲词典的覆盖度；另一个是因为只使用单词，有利于控制词语⁴的数量，并且也方便计算词语与词语之间的相似度。当然，这样只使用单词（会有一个将词语拆开得到若干单词的过程），会带来一定的歧义，在后面我们会利用一个特征模板来解决这个问题。

然后，我们用打分函数对词典中的每一个词汇都进行打分，并从中选择打分高的组成我们最终的种子词典。具体的，我们把上述种子词典放到基准语义解析器中，并定义若干特征来衡量词典中词汇的质量，模型中某个词汇的词汇质量特征所对应的参数值就是该词汇的质量度量，参数值越高，表示该词汇越准确，越有用，相反，参数值越低，表示该词汇越不准确。这是因为我们使用对数线性 (log-linear) 模型来建模语义解析过程，对于解析过程中某个词汇的使用能够导致最后生成准确的逻辑表达式，那么该词汇的词汇质量特征所对应的参数值就会更新（值变大）。这样一来，基准语义解析器训练完之后，词典中每一个词汇都会有相应的打分（词汇质量特征所对应的参数值），打分越高，词汇的质量越好。对于每一个谓词，我们选择 top K (K=5) 的词汇组成我们的最终种子词典。实际上，为每个词汇都打上分数是很有必要的，图3.7展示了利用打分的种子词汇和不利用打分的种子词汇分别进行图传播的例子，图中方框的节点是谓词，圆圈的节点是已经映射的词语，

⁴在后面我们同样会使用词语，而不是单词，由于我们已经声明只使用单词 (unigram)，所以在里词语就是单词。

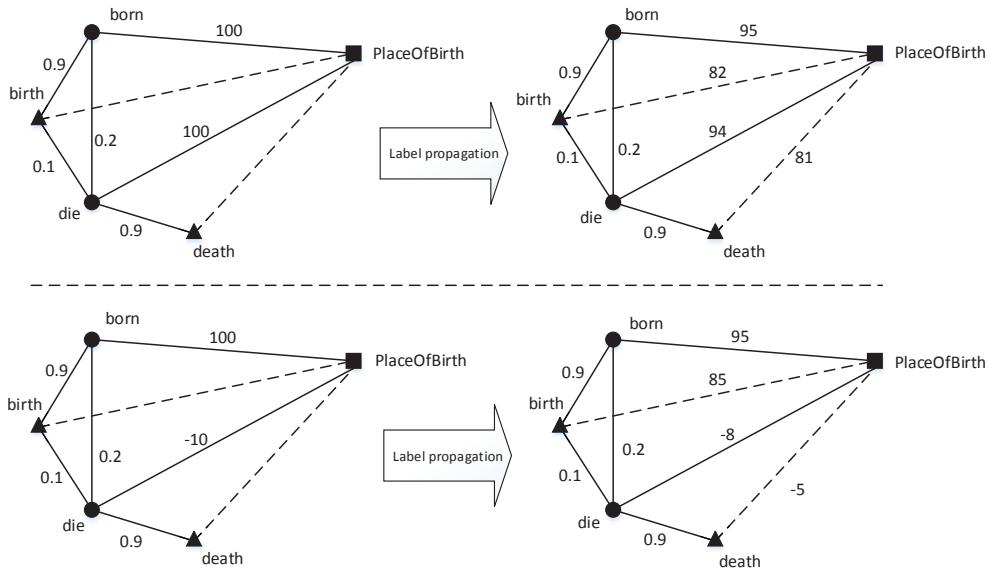


图 3.7: 对比有利用打分的种子词汇和没有打分的种子词汇分别进行图传播的例子

三角形的节点是未知映射的词语，词语与词语之间的边表示相似性度量，词语与谓词之间的实线表示已经的映射，虚线表示图传播算法以后学习到的映射。从例子中我们可以看到，如果我们不利用打分函数，而是对种子词典中的每一个词汇均匀对待，由于原种子词典中词汇的质量参差不齐，很多词汇是错误的，这样直接进行后面的图传播，会学习到质量更加无法得到保证的词汇，如词语“death”到 PlaceOfBirth 的映射，上部分的例子中不利用打分的，最后该词汇的得分有 81，下部分的例子中利用了打分函数，该词汇的最终得分是-5，显然后者是我们所希望的。

3.3.1.2 图构建

这一节我们要构建一个用于图传播的图，该图要包括已知的种子词典，其中每个词汇的词语端对应到一个节点，谓词端对应到一个标签（label），种子词典所对应的节点，我们称为标注好的节点（labeled nodes）；图中还包括出现在语料中的需要为之学习新映射的词语，这部分词语所对应的节点，我们称为未标注好的节点（unlabeled nodes）；另外我们在图中还附加的桥接节点（bridge⁵ nodes），这部分节点同时与标注好的节点和未标注好的节点相连，充当了从标注好的节点传播到未标注好的节点的桥梁。图3.8展示了我们所要构建的图的大概样式。图中的词汇并不是显式的，而是以这样一种形式：每一个节点（词语）在每一个标签（谓词）上都有一个概率值（对应词语到谓词映射的分数），那么现在我们已知了标注好的

⁵这里的 bridge 与前面介绍的用来充当词汇功能的 bridge 方法不是一个东西，只是在功能上都充当了桥连接的功能，所以都称为 bridge。

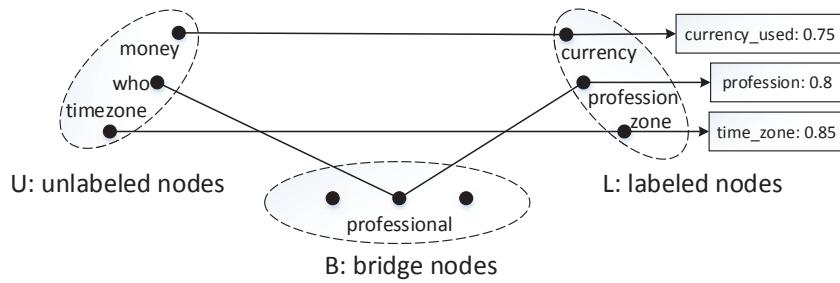


图 3.8: 用于图传播的图

节点在标签上的分布，我们需要构建词语与词语之间的相似性度量来学习未标注好的节点在标签上的分布。

我们有利用三种资源来构建词语与词语之间的相似性度量。

我们使用的第一种资源是词语的词向量表示 (word embedding)。近年来，词向量表示 [103, 104, 102] 在自然语言处理领域运用非常广泛，尤其是在跟语义相似度相关的任务中 [36]，词向量表示都起到了很多的作用。我们这里的目的是，在未标注好的词语中找到意思相近的标注好的词语，并把标注好的词语的标签赋给未标注好的词语。我们利用公开的词向量表示 Huang et al. [65]，并使用余弦相似度来计算两个词语之间的相似度。

第二种资源是复述表中抽取出来的复述词语对。这部分我们想利用复述词语对，如 “*money*” 和 “*currency*”。我们从 Paralex 语料 [47] 中构建这种复述词语对。Paralex 语料是一个大型的包含复述问句的语料，该语料是从 WikiAnswers⁶上收集来的，语料中包含问句-簇，每个簇中的问句都表达相同的意思（由用户自主标注）。我们使用标准的机器翻译里面的对齐算法从 Paralex 语料中抽取出词语对齐，我们再从这些对齐中抽取出单词 (unigram) 来构成复述词语对的表，表中每一个复述词语对都可以利用对齐的分数作为该词语对相似性的度量。这个复述词语对的表也是我们想要利用的，因为它的来源是复述问句集合。

第三种资源是 PPDB 中的词语表。具体的，我们使用的是 PPDB-2.0[117] 来计算词语之间的相似性。由于我们只使用单词 (unigram)，所以我们利用 PPDB-2.0 中的词典 (lexical) 部分，并直接利用它们的分数来作为词语之间的相似性值。另外我们对 PPDB-2.0 中词典部分中的词语都作取词干处理。PPDB-2.0 中对于词语之间的相似性度量是综合考虑了很多因素的结果，表3.1展示了我们利用 PPDB-2.0 中词语之间的相似性度量得到的与词语 “*currency*” 最相似的若干词语。

实际上，我们还可以利用其它的词典资源，如利用 WordNet⁷中的同义词集，并且已经利用 VerbNet 来学习高覆盖度的词汇。然而我们发现利用同义词集得到的相似词语基本都被我们利用前面三种资源所得到的词语所覆盖掉，所以这里，

⁶<http://www.answers.com/Q/>

⁷<https://wordnet.princeton.edu/>

neighbor words	scores
<i>guilder</i>	3.05
<i>coin</i>	3.03
<i>taxa</i>	3.00
<i>les</i>	2.89
<i>exchange</i>	2.85
<i>monetary</i>	2.76
<i>money</i>	2.63

表 3.1: PPDB-2.0 中与词语 “*currency*” 最相似的几个词语

我们并不利用 WordNet。Wang et al. [135]充分利用词型变化（如动词的不同形态）来建模词典。他们的词典是基于 CCG 的，他们为一个词语的不同形态设计句法部分不同，而语义部分相同的词汇模板；而对于同一个时态的不同词语，就会具有相同的句法部分和不同的语义部分。基于这种形式，他们得到了质量更高的词典。我们这里使用的是词语的词根，所以可以不考虑词型的变化。

为了进一步限制图的大小（节点的数目），我们为每一个未标注的节点选择 top 10 最相似的标注好的节点，并选择 top 5 最相似的桥节点；对于每一个桥节点，我们选择 top 5 的最相似的标注好的节点。另外，我们还考虑标注好的节点之间的连接。

最后，两个词语 w_1 和 w_2 之间的总的相似性是综合利用三类相似性度量方法计算而来，计算公式如下：

$$\begin{aligned} sim(w_1, w_2) = & \alpha sim_1(w_1, w_2) + \beta sim_2(w_1, w_2) \\ & + (1 - \alpha - \beta) sim_3(w_1, w_2) \end{aligned} \quad (3.1)$$

上述公式中， sim_1 是利用词向量表示得到的相似度， sim_2 是利用复述词语对得到的相似度， sim_3 是利用 PPDB 中的词语表得到的相似度。在最终的计算之前，每一部分的相似性度量都经过了归一化处理。上述公式中的超参数都在开发集上进行调试所得。

3.3.1.3 图传播

我们利用图传播算法来把标注好的节点的标签信息依循图结构传播给未标注的节点。这种标签传播算法基于如下假设：图中相似的节点应该具有相似的标签。前面已经介绍过，标签对应知识库本体中的谓词，节点对应到词语，为节点赋予标签，就是得到一个词汇，赋予标签的概率值可以作为该词汇的一个打分。在这节的工作中，我们利用 modified Adsorption algorithm[131]。该算法要得到未标注节点在标签上的分布 \hat{Y}_v ，该分布满足如下约束：

$$\min_{\widehat{Y}} \quad \mu_1 \sum_{v \in V_L} p_1 \|Y_v - \widehat{Y}_v\|_2^2 + \quad (3.2)$$

$$\mu_2 \sum_{v,u} p_2 W_{v,u} \|\widehat{Y}_v - \widehat{Y}_u\|_2^2 + \quad (3.3)$$

$$\mu_3 \sum_v p_3 \|\widehat{Y}_v - R_u\|_2^2 \quad (3.4)$$

上述目标函数中包括三个部分：第一部分是为了使得种子节点的标签分布 Y_v 尽量保持不变；第二部分是为了保证平滑度，也就是相似的节点具有相似的标签；第三部分是为了保证未标注节点的标签分布符合一个均匀分布 R_u 。 μ_1 、 μ_2 和 μ_3 是超参数，是对于各部分的侧重程度的一个调节，我们同样在开发集上对该超参数进行调试。我们使用 Junto 标签传播算法工具包⁸来进行标签传播。

3.3.2 基于扩充词典的语义解析

图传播算法结束以后，每一个未标注的词语都在标签上有一个概率分布。每一组（词语，标签）就是一个词汇。这些词汇与种子词汇一起构成我们的新词典。我们使用 SEMPRE[16, 15] 系统作为我们的基准语义解析器。我们把新词典作为语义解析器的初始词典，并在语料上进行语义解析的训练和测试。在这里我们同样利用 lemmaAndBinary 特征，前面已经证明该特征十分有效。

3.4 实验

我们为两种词典扩充方法都分别设计了实验来论证方法的有效性。两个方法的实验都在两个公开数据集上都进行了对比实验：WEBQUESTIONS 和 FREE917。

WEBQUESTIONS 数据集是由 Berant et al. [16] 收集而来。该数据集总共有 5800 个样例，每个样例由问句及其答案组成。按照标准，3778 个样例用于训练，2032 个样例用于测试。

FREE917 数据集有 Cai and Yates [22] 收集而来。该数据集总共有 917 个样例，每个样例由问句及其逻辑表达式组成。按照标准，641 个样例用于训练，276 个样例用于测试。

两个数据集都使用 Freebase 作为知识本体。

⁸<https://github.com/parthatalukdar/junto>

System	Prec.	Rec.	F1 (avg)
都不使用 FT_1 、 FT_2	41.6	45.4	39.5
使用 FT_1 , 不使用 FT_2	48.0	51.3	44.7
使用 FT_2 , 不使用 FT_1	46.3	52.4	45.5

表 3.2: 在 WEBQUESTIONS 数据集上的 A 组实验

System	Prec.	Rec.	F1 (avg)
都不使用 FT_1 、 FT_2	61.9	61.9	61.3
使用 FT_1 , 不使用 FT_2	65.1	66.0	65.1
使用 FT_2 , 不使用 FT_1	66.6	66.0	65.9

表 3.3: 在 FREE917 数据集上的 A 组实验

3.4.1 基于桥连接的词典学习的语义解析的实验

3.4.1.1 实验设置

在两个数据集上, 我们都采用标准的训练/测试数据分割, 若进行开发测试, 则将训练数据按 0.8/0.2 的比例切分, 其中 80% 的数据用来训练, 剩下 20% 的数据用来测试。系统评价指标同样采用公认的标准 [16]: 平均准确率。我们的实验都是基于 SEMPRE 系统完成。我们所使用的解析算法(除了桥连接部分)、模型的其它参数和学习算法都与基准系统相同。训练的迭代次数分别是 3 和 6。在第一次迭代训练的时候开启桥连接功能, 在其它训练轮次和测试环节关闭桥连接功能。用于获取核心词汇的阈值 K 都设置为 5。

3.4.1.2 实验结果与分析

对比不同的特征模板 (A 组实验) 首先, 我们来对比我们提出的新的特征模板-lemmaAndBinary(记为 FT_2)和 Berant and Liang [15]提出的特征模板-lemmaAndBridge(记为 FT_1)。由于 lemmaAndBridge 仅在使用桥连接的情况下生效, 所以我们在基准系统上面验证两个特征模板的效果(都开启桥连接功能)。我们都在开发集上进行测试, 在两个数据集上的结果如表3.9和表3.10所示。从结果我们可以看到, 特征模板 FT_1 和 FT_2 都非常有效, F1 值都提升了 4%-5%, FT_2 比 FT_1 更加有效。实际上, FT_2 特征模板是包含 FT_1 特征模板的。在后面的实验中我们都在我们的系统上使用 lemmaAndBinary 特征模板。

检验获取核心词典的作用 (B 组实验) 为了验证获取核心词典的作用, 我们对比了不获取核心词典而直接使用整个新词典的系统和使用获取到的核心词典的系统。

System	Prec.	Rec.	F1 (avg)
原始词典 (L_o)	48.6	57.5	48.6
基于 L_o 的核心词典	49.2	54.8	48.5
新词典 (L_n)	50.1	58.0	49.7
基于 L_n 的核心词典	51.5	56.8	50.4

表 3.4: 在 WEBQUESTIONS 数据集上的 B 组实验

System	Prec.	Rec.	F1 (avg)
原始词典 (L_o)	63.9	63.9	63.3
基于 L_o 的核心词典	64.5	62.8	64.1
新词典 (L_n)	66.9	69.2	66.4
基于 L_n 的核心词典	67.5	68.3	67.0

表 3.5: 在 FREE917 数据集上的 B 组实验

我们都在开发集上进行测试，实验中都使用 lemmaAndBinary 特征模板，关闭桥连接功能，此外我们还在原始词典的基础上测试了获取核心词典的效果，在两个数据集上的结果如表3.4和表3.5所示。从结果我们可以看到，基于投票机制的核心词典获取方法可以提高系统的准确率，虽然会牺牲一点召回率，但是对系统整体性能是起到提高作用的。另外我们还发现获取核心词典的方法能够大大压缩词典的大小。系统所利用的核心词汇是很少的，只占用了原始词汇的 1/500 不到（在 WEBQUESTIONS 数据集，原始词汇中类别词汇和关系词汇的数量分别是：282005 和 132111，而核心词汇的数量分别是：66 和 335）。在后面的实验中，我们都在我们的系统上使用基于投票机制的核心词典获取方法。

主要结果(C 组实验) 我们在我们最终的系统中使用了新的特征模板-lemmaAndBinary 和获取核心词汇的策略，并与基准系统在测试集上进行对比实验，在基准系统上，我们开启了桥连接功能，并选择使用了 lemmaAndBridge 特征模板，在我们的系统上，我们使用了 lemmaAndBinary 特征，并同时使用基于投票机制的核心词典获取方法，并设置实验再次测试这两种策略带来的效果。在两个数据集上的结果如表3.6和表3.7所示。从结果我们可以看到，基于桥连接的词典学习方法在使用 lemmaAndBinary 特征模板和基于投票机制的核心词典获取的情况下，比桥连接方法更有效。更重要的是，我们的方法能够学习到新的词汇，可提供后续分析和使用。表3.8展示了几个我们新学到的词汇。这些词汇都是原始词典中不存在的词汇，而是在训练过程中，基于桥连接方法，自动引入的词汇，并加以学习而得来的，我们可以看到这些词汇也符合我们的直觉，是我们所需要的，有了这些新词汇，能够有

System	Prec.	Rec.	F1 (avg)
基准系统	49.7	58.2	49.6
我们的系统	51.3	59.3	50.9
我们的系统 - FT_2	48.9	55.5	46.7
我们的系统 - 核心词典	50.2	59.8	50.0

表 3.6: 在 WEBQUESTIONS 数据集上的 C 组实验

System	Prec.	Rec.	F1 (avg)
基准系统	65.1	66.0	65.1
我们的系统	68.3	69.0	67.6
我们的系统 - FT_2	65.4	63.3	64.6
我们的系统 - 核心词典	66.8	70.3	66.0

表 3.7: 在 FREE917 数据集上的 C 组实验

效提高语义解析器的性能。

3.4.2 基于半监督的词典学习的语义解析的实验

3.4.2.1 实验设置

依然在数据集 WEBQUESTIONS 和 FREE917 进行测试，训练/测试数据分割还是采用标准形式。训练迭代次数依然设置为 3 次和 6 次，柱搜索的搜索空间设置为 200 和 500。在相似性计算中，我们设置 $\alpha = 0.05$, $\beta = 0.85$ 。在 Junto 工具包中我们设置： $\mu_1 = 0.55$, $\mu_2 = 0.44$, $\mu_3 = 0.01$, $\beta = 2$ 。

为了验证我们方法的有效性，我们做了如下三种设置：在基准语义解析系统中关闭桥连接功能，使用原始词典，该系统记为 Base；在基准语义解析系统中开启桥连接功能，使用原始词典，该系统记为 Base+bridge；在基准语义解析系统中关闭桥连接功能，使用我们新的词典，该系统记为 Our approach。另外我们还和其他系统进行了对比。

词语	谓词
<i>college</i>	<i>Education</i>
<i>wife</i>	<i>Spouse</i>
<i>kill</i>	<i>cause_of_Death</i>
<i>win</i>	<i>champion</i>

表 3.8: 若干复述模板对的例子

System	Prec.	Rec.	F1 (avg)
Berant et al. [16]	48.0	41.3	35.7
Yao and Van Durme [148]	51.7	45.8	33.0
Berant and Liang [14]	40.5	46.6	39.9
Bao et al. [11]	—	—	37.5
Bordes et al. [19]	—	—	39.2
Yang et al. [146]	—	—	41.3
Bast and Haussmann [12]	49.8	60.4	49.4
Yao [147]	52.6	54.5	44.3
Berant and Liang [15]	50.5	55.7	49.7
Yih et al. [149]	52.8	60.7	52.5
Base	51.0	47.6	40.5
Base + bridge	50.0	58.5	50.0
Our approach	51.6	59.7	51.2

表 3.9: 在 WEBQUESTIONS 数据集上的结果对比

System	Prec.	Rec.	F1
Cai and Yates [22]	67.0	59.0	63.0
Kwiatkowski et al. [87]	76.7	68.0	72.1
Bast and Haussmann [12]	72.0	67.8	69.8
Base	71.2	59.5	64.8
Base + bridge	69.4	64.4	66.8
Our approach	71.5	67.9	69.6

表 3.10: 在 FREE917 数据集上的结果对比

3.4.2.2 实验结果与分析

表3.9和表3.10分别展示了在两个数据集上，我们的方法与其他系统方法的结果对比。从表中我们可以看到：

1. 我们的方法取得了相当可观的结果：我们的系统远远超过了基准系统(Base)，在 WEBQUESTIONS 数据集上提高了 9.7 个点，在 WEBQUESTIONS 数据集上提高了 4.8 个点；相比于加桥连接方法的基准系统，我们的系统同样超过了它 (Base+bridge)。
2. 我们学到的新词典的覆盖度远远超过了原词典的覆盖度，这点从系统的召回率上可以看出一个大概，召回率高的系统，一般词典的覆盖度更大。通过利用大量容易获取的文本语料和词典资源，我们的方法能够提高词典的覆盖

度，从而提升语义解析器的性能。在 WEBQUESTIONS 数据集上，我们的方法取得了几乎最高的召回率 (67.9 VS 68.0)，在 WEBQUESTIONS 数据集上表现也很不错，排名第三。在两个数据机上，相比基准系统，召回率都提升非常大。

3. Berant et al. [16]所提出的桥连接方法 (bridge) 十分有效，前面已经分析过，桥连接方法充当了词汇的功能，但是它并没有学习到新的词汇。我们的方法不仅学习到了新的词汇，且新词汇的覆盖度也超过了使用桥连接方法的系统。

对比我们的方法与当前最好的方法 我们的系统虽然比基准系统提升很大，但是比当前最好的系统的效果还是要差一些的。在 WEBQUESTIONS 数据集上，Yih et al. [149]在 F1 值上取得了 52.5 的结果，我们的方法是 51.2。他们的方法在准确度上和召回率上效果都不错。他们的方法并不是基于传统组合语义的，他们并不需要词典和文法，他们也不是自底向上的进行语义解析的，他们把语义解析分为 3 步：第一步进行实体链接；第二步进行关系匹配；第三步利用规则添加附属条件。我们是先有词典再进行解析，也就是语义落地的步骤先完成（词典构建），再进行结构的预测（用文法进行组合）。而他们是在解析的步骤中进行语义落地，这样以来他们并不受限与词典，并且在语义落地的时候可以利用判别式的方法，虽然搜索空间会有点大，但是在利用一些启发式规则的前提下，可以有效提升系统的性能。虽然，我们提出来的基于词典的语义解析方法现在表现比不过不使用词典的语义解析方法，但是我们认为基于词典的语义解析方法依然有它的优势，一方面，它所学习到的词汇应用也非常广泛；另一方面，基于词典的语义解析方法在解析的时候，搜索空间完全可以由词典大小和文法规规模来确定，而不使用词典的方法的搜索空间往往由知识库的规模决定。

使用单词作为词典的效果 在前面已经介绍过，词汇中我们只使用单词，不使用多词。一方面是为了控制图的大小，另一方面是为了方便计算单词之间的相似度。为了论证相比原来的词典，我们这种只用切分后的单词作为词汇的方法并不会影响语义解析的性能。我们在两个数据及上进行了对比实验，一种是使用原来的词典，一种是使用切分原来词典中的多词，只使用单词的词典，这里都是原始词典，没有利用图传播算法学习新词典。对比结果如表3.11和表3.12所示，从表中，我们可以看到这两中词典的效果差不多，甚至只使用单词的词典的效果还好一点点，这也表明，我们可以为了后面的方便，放心的使用只有单词的词典。

不同资源对于词典学习的效果 我们在计算词语的语义相似性的时候，有使用到三种语料资源，为了比较这三种资源的效果，我们设置了对比实验，4 个对比系统设置如下：只使用词向量表示；只使用复述词语对；只使用 PPDB-2.0；同时使用 3 中资源。我们使用上述设置分别计算词语之间的相似度，再分别学习词典，并用于语义解析中。最终语义解析的结果如表3.13所示，从表中我们可以看到：

System	Prec.	Rec.	F1 (avg)
Original seed	40.6	47.5	40.6
New seed	51.0	47.6	40.5

表 3.11: 在 WEBQUESTIONS 上使用不同词典的结果对比

System	Prec.	Rec.	F1 (avg)
Original seed	69.8	59.0	63.9
New seed	71.2	59.5	64.8

表 3.12: 在 FREE917 上使用不同词典的结果对比

1. 仅仅使用词向量作为语义相似性的度量所得到的新词典的效果并不理想。我们认为词向量中语义相似性中的语义其实考虑了很多的方面，而我们需要的是具有相同意思或相似意思的方面。比如，利用词向量来计算语义相似度，“apple” 和 “orange” 肯定具有相当高的相似度，但我们不太想要这种相似度。
2. 仅仅使用复述词语对的效果也不是很明显，相比基准系统，该方法的提升也只有一点点。我们认为这是由于在计算复述词语对之间的相似性的时候，我们只是利用了机器翻译里面的对齐算法。这种对齐里面存在相当多的噪音。
3. 仅仅使用 PPDB-2.0 的效果十分的好。我们认为主要是因为 PPDB-2.0 里面的同义词词语集合都是从复述语料中抽取出来的，相比仅使用对齐的方法，PPDB-2.0 里面还考虑了很多其他因素，使得得到的词语大体符合复述的要求。
4. 同时使用三种语料进行语义相似性度量计算的效果最好，我们认为这三种语料资源的作用可以相互补充。

学习到的新词汇 我们的目的是为语义解析学习到覆盖度足够的词典。我们的方法利用容易获取的文本和词典资源来从种子词典中学习到覆盖度更广的新词典。表3.14展示了在 WEBQUESTIONS 数据集上，我们方法所学到的几个新的词汇。表中 predicate 表示词汇中的谓词，seed phrase 表示种子词汇中的词语，learned phrase 表示新学习的的词汇中的词语，score 表示该词汇的词汇质量打分。从表中我们可以看到，我们的方法可以利用种子词汇和词语之间的相似性学习到了新的词汇。

简单的错误分析 我们对实验结果中错误的样例进行了简单的分析，我们发现词典覆盖度不够的问题基本解决，剩下一些复杂的难处理的问题，比如多元谓词、从句、最高级等，这些语言现象其实很常见，但现阶段面向开放域的的语义解析还很难处理这类复杂的情形，这需要设计足够准确和足够覆盖度的文法来表示这种语言现象。

System	Prec.	Rec.	F1 (avg)
Base	40.6	47.5	40.6
Only word vector	41.8	45.9	39.7
Only paraphrase table	50.0	46.3	39.4
Only PPDB-2.0	51.2	58.7	50.6
All	51.6	59.7	51.2

表 3.13: 在 WEBQUESTIONS 数据集上, 使用不同资源进行相似性度量后再进行词典学习的效果

Predicate	Seed phrase	Learned phrase	score
currency	<i>currency</i>	<i>money</i>	4.91
Education	<i>education</i>	<i>school</i>	4.75
Religion	<i>religion</i>	<i>believe</i>	2.30
Profession	<i>professional</i>	<i>who</i>	2.30

表 3.14: 在 WEBQUESTIONS 数据集上, 我们方法所学到的几个新的词汇, 以及该词汇的词汇质量打分

3.5 本章小结

本章是从语义落地的角度研究语义解析。词典是基于组合语义的传统语义解析方法的核心, 长期以来, 困扰研究者的就是词典覆盖度不够的问题。为了解决这个问题, 我们在现有工作的基础上, 提出了一种基于桥连接的词典学习的语义解析方法。该方法在训练的时候, 可以引入新的词汇, 并加以学习。为了进一步提高词典的质量, 为了提高词汇的准确度, 我们还设计了新的词语-二元谓词的模板。为了压缩词典的大小, 获取到核心词典, 本文还提出了基于投票机制的核心词典获取方法。实验结果表明, 我们的方法相比基准系统有很大的提升。

为了进一步解决语义解析中词典覆盖度不够的问题, 我们提出了基于半监督词典学习的语义解析方法。与现有方法对比, 该方法不仅仅利用知识库与文本之间的实体对的共现来学习词汇, 还充分利用文本与文本之间的语义联系来进一步扩充词汇。具体的, 我们首先利用知识库与文本之间的实体对共现来学习原始词典, 还利用已有语义解析器对每一个词汇进行打分, 并从中选取打分高的词汇, 组成我们的原始词典; 接着我们利用文本语料和词汇资源学习词与词之间的相似性; 最后, 我们利用图传播算法, 以及词与词之间的相似性, 从种子词汇中学习出新的词汇。我们把新的词典放到已有语义解析器中, 并在两个公开数据集上进行验证, 实验结果验证了我们方法的有效性: 扩充以后的词典的覆盖度显著提高。

本章的相关研究成果有两个: “基于桥连接的词典学习方法的语义解析”已被

《中文信息学报》录用; “**Semi-supervised Lexicon Learning from Unlabeled Corpora for Wide-Coverage Semantic Parsing**” 已投稿 Coling-2018 (长文)。

第四章 基于句子重写的语义解析

4.1 本章引论

语义解析首先也是一个结构预测的问题。对于一个简单的含有主谓宾的句子，语义解析要识别句子里面的主语、谓语和宾语，并与物理世界中实体和关系对应（主语和宾语都对应到实体，谓语对应到关系），同时还要预测出主语与宾语之间的结构关系：哪个是施事着，哪个是受事着，施事着对受事着做了什么，或者存在什么关系。前者属于语义落地，后者属于结构预测。图4.1中展示了一个语义解析的例子，例子中“*German*”是一个实体，“*what*”对应到一个变量实体，它们之间的关系是“*the capital of*”。

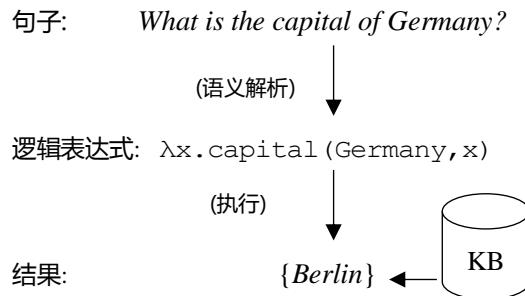


图 4.1: 语义解析的一个例子

语义解析中的结构映射是指源端句子的结构到目标语义表示结构的映射。通常，目标端语义表示的结构是由知识库本体决定的，由于知识库本体一般是固定的，也意味着目标端对于每一个语义的表示都是固定的。然而自然语言表达极具丰富性和多样性，一个意思（语义）通常可以有多种表达，如询问姚明的年龄，可以有下面多种表达：“姚明多大了？”、“姚明什么年纪了？”、“姚明的年龄是多少？”、“姚明多大岁数？”等。对于句法结构差异不大，只是使用了不同的词汇，可以使用前述词典学习方法来解决。但是当句法结构发生变化，现有语义解析难以处理这种情况。如下面这个例子：

```
s1 What is the population of Berlin?  
s2 How many people live in Berlin?  
lf1 λx.population(Berlin, x)  
lf2 count(λx.person(x) ∧ live(x, Berlin))
```

例子中两个句子 s_1 和 s_2 具有一个意思，都是询问柏林的人口，显然这两个句子的句法结构截然不同。例子中逻辑表达式 lf_1 和 lf_2 是基于两种不同知识库本体对柏林的人口这个语义的表示。前面已经提到目标端知识库本体一般是固定的，也就是

(a) 之前方法的语义解析例子

```

s0: What is the name of Sonia Gandhi's daughter?  

l0: λx.child(Sonia Gandhi, x)  

r0: {Rahul Gandhi (Wrong answer), Priyanka Vadra}

```

(b) 我们方法的语义解析例子

```

s0: What is the name of Sonia Gandhi's daughter?  

s1: What is the name of Sonia Gandhi's female child?  

l1: λx.child(Sonia Gandhi, x) ∧ gender(x, female)  

r1: {Priyanka Vadra}

```

图 4.2: 对比之前的语义解析方法和我们的使用句子重写的语义解析方法

说实际操作中，我们只有 l_{f_1} 或者 l_{f_2} ，而在自然语言端， s_1 和 s_2 是可以同时存在的，并且这种现象是很普遍的。这样一来，就容易造成源端自然语言句子的结构与目标端知识库本体的结构不匹配。这种结构不匹配问题，也是语义解析所面临的一大挑战。再看上面的例子，若是把 s_1 映射到 l_{f_1} ，或把 s_2 映射到 l_{f_2} ，对于这种结构是一致的情况，现有语义解析可以取得可观的准确率，但要是目标语义表示与句子的结构不一致，如语义表示是 l_{f_1} ，句子是 s_2 ，或者语义表示是 l_{f_2} ，句子是 s_1 ，这样一来就会给语义解析造成很大的困难。

为了解决上述的自然语言句子的结构与知识库本体的结构不一致的问题，本章我们提出了基于句子重写的语义解析方法，对于这种结构不匹配问题，我们通过找到句子的另外一种表述（还是句子），该表述与原句子具有相同的意思，而该表述的结构与目标知识库本体的结构是一致的。图4.2中展示了我们使用句子重写的语义解析方法的例子，以及与之前的方法进行对比。表格中 s_0 是句子； l_0 是之前语义解析方法得到的与句子 s_0 对应的逻辑表达式（使用 lambda-表达式的方式）； r_0 是利用 l_0 从知识库中得到的结果； s_1 是我们利用句子重写方法从 s_0 得到的新句子； l_1 是我们得到的与句子 s_1 所对应的逻辑表达式； r_1 是利用 l_1 从知识库中得到的结果。结果 r_0 中“Rahul Gandhi”是一个错误的答案，因为他是“Sonia Gandhi”的儿子。例子中，我们需要把句子 s_0 : “What is the name of Sonia Gandhi's daughter?” 解析成 l_1 : $\lambda x.\text{child}(\text{S.G.}, x) \wedge \text{gender}(x, \text{female})$ ，之前的方法直接对 s_0 进行解析，往往得到的是 l_0 : $\lambda x.\text{child}(\text{S.G.}, x)$ ，这正是因为目标语义表示 l_1 与源端自然语言句子之间存在结构不匹配的问题：词语“daughter”对应到目标语义表示 $\text{child} \wedge \text{female}$ 。我们的方法通过句子重写的方法，由 s_0 得到新句子 s_1 ，具体的，把句子中“daughter”重写到“female child”，再对 s_1 进行解析，由于 s_1 的结构与目标语义表示 l_1 的结构是一致的，因此 s_1 到 l_1 的解析过程会比从 s_0 到

l_1 的解析过程更容易。

具体的，我们对结构不匹配问题进行了归纳和总结，有如下两类不匹配问题：

1. 1-N 型不匹配：句子中一个简单的词语对应到知识库本体中多个项。如“*daughter*”一词对应到知识库本体中的项为 `child&female`¹。
2. N-1 型不匹配：知识库本体中的一个项对应到句子中的多个词语，如知识库本体中的项 `population` 是由自然语言中“*how many people*”和“*live in*”共同表达的。

为了解决上述两类不匹配问题，本章我们分别提出了两种句子重写的方法：一种是基于字典替换的句子重写方法，用来解决 1-N 型不匹配问题，具体的，我们用字典中对名词的解释来替换该词，得到新的句子；另一种是基于模板替换的句子重写方法，用来解决 N-1 型不匹配问题，具体的，我们利用复述模板对句子进行重写。

对于给定的句子，我们利用句子重写会生成很多新的句子，并利用语义解析器对每一个新句子都进行解析，得到对应的逻辑表达式。另外，我们提出了一个打分排序函数，该打分函数同时考虑了句子重写部分的特征，和语义解析部分的特征，综合选择得到最优的逻辑表达式。

现阶段已经有一些工作开始关注语义解析中结构不匹配的问题。Kwiatkowski et al. [87]通过本体匹配的方法来解决结构不匹配问题，具体的，他们先生成句子的领域无关的字面意思表示，然后利用本体匹配把上述字面意思表示映射到目标本体表示。然而，他们的方法还是很难处理这种 1-N 型和 N-1 型不匹配问题。Berant and Liang [14]则是利用复述的方法，具体的，他们先利用规则从句子生成候选逻辑表达式，再利用规则从候选逻辑表达式生成新句子，最后把原句子与新句子构成一个复述对，再用一个复述模型来进行复述判断。但是他们的方法在生成候选逻辑表达式的时候，并没有去解决上述结构不匹配的问题。相比上述两种方法，我们的方法在解析前就尽量去消除这种结构不匹配，从而大大减轻了后面语义解析的负担。图4.3是对这些方法的简单对比。

使用句子重写技术来解决不匹配问题的这种思路已经在其他任务上得到了验证。为了解决机器翻译中源语言与目标语言之间词序不匹配的问题，特别是某些语种之间的翻译，如德语到英语的英语，词序的变化非常大，Collins et al. [35]提出了一种重新构建句子结构（restructuring）的方法，具体的，对于源语言端的句子，在句法树的基础上，利用若干条规则对句子的结构进行调整，从而得到新的句子，其目的是为了新的句子的词序与目标端的句子的词序尽量匹配。为了解决同声翻译中由源语言与目标语言的词序不同所带来的翻译延迟问题，He et al. [58]采用相

¹在本章中，我们会为了方便理解而简化逻辑表达式，例如使用 `female` 来表示 `gender.female`

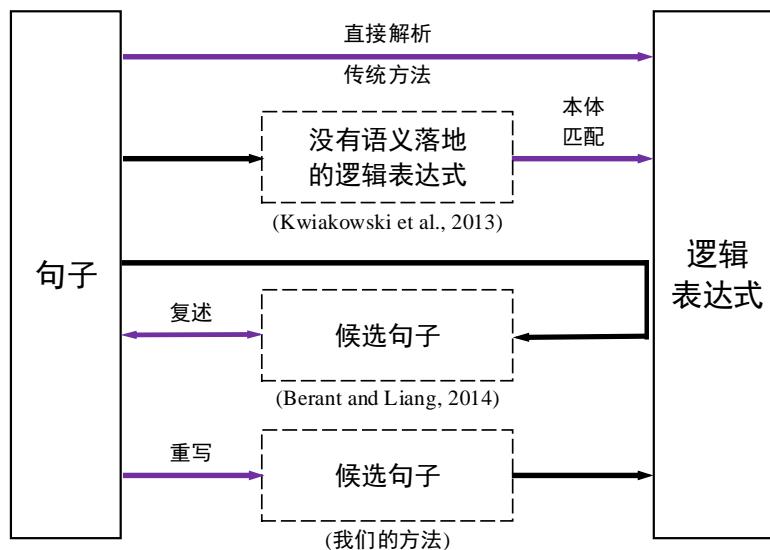


图 4.3: 针对结构不匹配问题, 我们的方法与其他方法的对比

同的思路, 在源语言端句子的依存树上, 利用规则对句子进行调整, 目的也是为了新的句子尽量接近目标端的句子的词序。

句子重写与复述生成是相同的任务, 都是给定一个句子, 生成新的句子, 其意思与原句子相同。Narayan et al. [111]在同一年独立地提出了基于复述生成的语义解析方法。沿着本文的研究思路, Dong et al. [45]提出基于复述的自动问答方法; Iyer et al. [66]在语义解析中利用复述的形式来扩充语料, 提高语义解析器的泛化性和鲁棒性。

我们在公开数据集 WEBQUESTIONS 上进行了实验: 实验结果表明我们的方法能够有效解决结构不匹配的问题, 能够有效提高语义解析器的性能。

本章工作的主要贡献可以总结为以下两点:

1. 针对语义解析中自然语言句子与目标知识库本体之间结构不匹配的问题, 我们提出了基于句子重写的语义解析方法。我们方法的出发点是当句子与知识库本体结构不匹配的时候, 句子总存在另外一种表达, 该表达与原句子有相同的意思, 但是新表达的结构与目标本体之间的结构是一致的。
2. 我们总结了两类结构不匹配的问题, 并分别提出了两种句子重写的方法, 针对 1-N 型不匹配问题, 我们提出了基于字典替换的句子重写方法; 针对 N-1 型不匹配问题, 我们提出了基于模板替换的句子重写方法。实验结果也论证了我们方法的有效性。

Word	Logical Form	Wiktionary Explanation
<i>son</i>	<i>child</i> ∧ <i>male</i>	<i>male child</i>
<i>actress</i>	<i>actor</i> ∧ <i>female</i>	<i>female actor</i>
<i>father</i>	<i>parent</i> ∧ <i>male</i>	<i>male parent</i>
<i>grandaprent</i>	<i>parent</i> ∧ <i>parent</i>	<i>parent of one's parent</i>
<i>brother</i>	<i>sibling</i> ∧ <i>male</i>	<i>male sibling</i>

表 4.1: 若干样例，每个样例包括一个词语，它的目标语义表示以及该词语在 Wiktionary 中的解释

4.2 句子重写

这一节，我们将介绍我们提出的句子重写的方法，分别是针对 1-N 型不匹配问题的基于字典替换的句子重写方法，和针对 N-1 型不匹配问题的基于模板替换的句子重写方法。

4.2.1 基于字典替换的句子重写

在 1-N 型不匹配的情形中，一个词语会对应到一个复合的语义表示，比如，词语 “*daughter*” 的目标语义表示是 *child* ∧ *female*（表4.1中有更多的例子）。

为了解决这类 1-N 型不匹配问题，我们把原词语 (“*daughter*”) 重写成新的表达 (“*female child*”), 该表达与目标语义表示的结构是一致的。在本章工作中，我们利用词语在字典中的解释对词语进行重写。这是因为每一个词语在字典中的定义都会使用详细而简单的解释，而这些解释的结构恰好与目标语义表示的结构是同构的。表4.1中展示的例子中词语与目标语义表示之间都存在不匹配问题，而这类不匹配问题，都可以利用词语在字典中的解释来解决。比如，词语 “*daughter*” 在 Wiktionary²中的解释是 “*female child*”，恰好与目标语义表示 *child* ∧ *female* 具有相同的结构。

大部分情况下，这类 1-N 型不匹配问题，都是由句子中的普通名词 (common nouns) 引起的。因此，我们为了稍微限制一下重写得到的新句子的规模，我们只对原句子中出现的普通名词用它们在字典中的解释来替代。而通常一个句子中普通名词的数量不会太多，因此我们得到的新句子的规模是可控的。对于由基于字典替换的重写方法得到的新句子，我们会定义一个打分函数，利用若干特征模板，来对它们进行打分排序，详情请见第4.3节。

表4.2中展示了一个基于字典替换的句子重写样例。例子中，原句子 s 有两个普通名词 (“*name*” 和 “*daughter*”), 因此我们会得到 3 个新的句子，分别是 r_1 、

²<https://www.wiktionary.org/>

<i>s: What is the name of Sonia Gandhi's daughter?</i>
<i>r₁: What is the reputation of Sonia Gandhi's daughter?</i>
<i>r₂: What is the name of Sonia Gandhi's female child?</i>
<i>r₃: What is the reputation of Sonia Gandhi's female child?</i>

表 4.2: 基于字典替换的句子重写方法的样例

r_2 和 r_3 。在这三个候选的新句子中，第二个候选的句子 r_2 使我们想要的，因为它的结构与目标语义表示的结构是一致的，并且不会因为使用了其他词语（如，用“reputation”来替代“name”）的替换而带来额外的噪音。

在本章中，我们使用 Wiktionary 作为我们的字典来对句子中的普通名字进行重写。具体的，给定一个普通名词，我们使用在 Wiktionary 中该词语的解释中“Translations”一栏的解释。由于大部分的 1-N 型不匹配问题都是由普通名词引起的，因此我们只收集普通名词的解释。若一个词语在字典中存在多个解释，我们使用常见的解释（第一个）。另外，我们只使用解释中词语长度不超过 5 的解释。

4.2.2 基于模板替换的句子重写

在 N-1 型不匹配问题的情形中，一个复杂的自然语言表达对应到一个简单的目标语义表示。比如，考虑下面这个例子，我们需要把句子 s 映射到语义表示（逻辑表达式） lf ，该语义表示是以 Freebase 作为知识本体。

$$\begin{aligned} s: & How\ many\ people\ live\ in\ Berlin? \\ lf: & \lambda x. population(Berlin, x) \end{aligned}$$

例子中有三个词语：“how many” (count)、“people” (people) 和 “live in” (live) 会一起映射到二元谓词 population。表4.3中展示了更多的类似的例子。

Expression	Logical constant
how many, people, live in	population
how many, people, visit, annually	annual-visit
what money, use	currency
what school, go to	education
what language, speak, officially	official-language

表 4.3: 若干 N-1 型不匹配的例子

为了解决这种 N-1 型不匹配问题，我们提出一种基于模板替换的句子重写方法，该方法能够把这种具有复杂结构的句子转换成简单结构的句子。具体的，我们

模板 1	模板 2
<i>How many people live in \$y</i>	<i>What is the population of \$y</i>
<i>What money in \$y is used</i>	<i>What is the currency of \$y</i>
<i>What school did \$y go to</i>	<i>What is the education of \$y</i>
<i>What language does \$y speak officially</i>	<i>What is the official language of \$y</i>

表 4.4: 若干复述模板对的例子

利用复述模板对的集合 $P = \{(t_{i1}, t_{i2})|i = 1, 2, \dots, n\}$ 来重写句子，其中 t 是一个句子的模板，该模板有一个参数槽 $\$y$ ，参数槽对应一个实体， t_{i1} 和 t_{i2} 组成一个复述对。在本章的工作中，我们只使用只有一个参数槽的模板，表4.4中展示了若干复述模板对。

给定复述模板对数据库和一个句子 s_1 ，我们基于模板替换的句子重写方法将按照如下几个步骤对句子进行重写。

- 首先我们从句子 s_1 生成一组候选的模板 $ST = \{st_1, st_2, \dots, st_n\}$ ，具体的，我们对句子中的实体用槽 “\$y” 进行替代。比如，从句子 “*How many people live in Berlin*”，我们可以得到模板 “*How many people live in \$y*”。
- 接着，我们从复述模板对数据库中检索出所有的复述模板对 (t_1, t_2) ，其中 $t_1 \in ST$ 。比如，我们利用第一步中的模板 “*How many people live in \$y*”，可以检索得到这样的复述模板对 (“*How many people live there in \$y*”, “*What is the population of \$y*”), 其中第二部分就是 t_2 。
- 最后，我们把模板 t_2 中的槽 “\$y” 用第一步中的实体替换，得到新的句子 s_2 ，这个新句子就是我们利用基于模板替换的方法得到的一个重写的句子。比如，我们在 t_2 用 “*Berlin*” 替换 “\$y”，可以得到 “*What is the population of Berlin*”。通过这种方法，我们可以得到结构简单的，同时与目标语义表示 ($\text{population}(\text{Berlin})$) 的结构是一致的新句子。

为了控制通过基于模板替换方法得到的重写的新句子的数量，同时为了衡量使用一个复述模板对进行句子重写的质量，我们定义了若干特征来对这种重写进行打分排序（详情请见第4.3节）。

为了构建复述模板对数据库，我们使用类似于 Fader et al. [48]的方法来自动收集复述模板对。具体的，我们使用 WikiAnswers 复述语料库 [47]，该复述语料库包含 2 千 3 百万问句-簇 (question-clusters)，并且每个簇中的问题都表达了同样的意思 (都是用户自发标记的可合并的问题)。表4.5中展示了该复述语料库中的两个问句-簇。为了构建复述模板对，对于每一个簇，我们首先对于该簇中的问句

<i>How many people live in chembakolli?</i>
<i>How many people is in chembakolli?</i>
<i>How many people live in chembakolli india?</i>
<i>How many people live there chembakolli?</i>
<i>How many people live there in chembakolli?</i>
<i>What is the population of Chembakolli india?</i>
<i>What currency is used on St Lucia?</i>
<i>What is st lucia money?</i>
<i>What is the money used in st lucia?</i>
<i>What kind of money did st lucia have?</i>
<i>What money do st Lucia use?</i>
<i>Which money is used in St Lucia?</i>

表 4.5: 来自 WikiAnswers 复述语料库中的两个问句-簇

所共有的实体，用槽 “\$y” 进行替代，这样一来该簇中含有槽 “\$y” 的模板两两之间构成一个复述模板对。复述模板对中位置是有顺序的，也就是说，复述模板对 (t_1, t_2) 和复述模板对 (t_2, t_1) 是两个不同的复述模板对。为了保障复述模板对的有效性，我们只保留出现次数超过 3 的复述模板对。

4.3 基于句子重写的语义解析

在这一节，我们将介绍我们的基于句子重写的语义解析系统。图4.4是我们基于句子重写的语义解析方法的框架图。给定一个句子，我们先使用句子重写的方法对原句子进行重写，得到若干新的句子；接着我们利用基准语义解析器对每一个新的句子进行语义解析得到各自的候选逻辑表达式；最后我们对每一个逻辑表达式进行打分并排序，选择打分最高的作为我们的最终结果。接下来，我们将介绍我们所使用的基准语义解析器，以及我们所使用的打分函数。

4.3.1 基准语义解析器

在本章的工作中，我们使用基于议程（agenda-based）的语义解析器 [15] 作为我们的基准语义解析器，并用它来生成句子的逻辑表达式。该语义解析器也是基于 lambda-DCS 文法 [91] 的。解析的时候，我们使用 Berant et al. [16] 所公布的词典和文法，其中词典用来触发生成一元谓词和二元谓词，文法用来组合谓词生成最终的逻辑表达式。有点不同的地方是，我们新加了一条组合规则，用来生成稍微复杂一点的逻辑表达式（包含两个二元谓词），如 `child.obama^gender.female`。

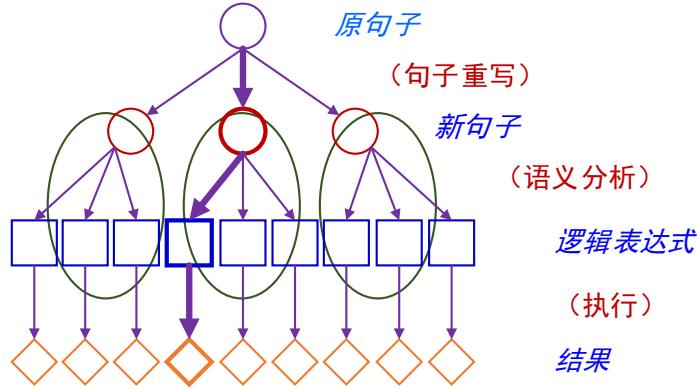


图 4.4: 我们提出的基于句子重写的语义解析方法的框架图

对于语义解析部分的模型学习，基准语义解析器学习如下策略函数：

$$p_\theta(a|s) = \frac{\exp\{\phi(a)^T \theta\}}{\sum_{a' \in A} \exp\{\phi(a')^T \theta\}} \quad (4.1)$$

其中 a 是一个策略， s 是解析过程中的状态 (state)。策略函数的参数 θ 更新如下：

$$\theta \leftarrow \theta + \eta R(h_{target}) \sum_{t=1}^T \delta(h_{target}) \quad (4.2)$$

$$\begin{aligned} \delta_t(h) &= \nabla_\theta \log p_\theta(a_t|s_t) \\ &= \phi(a_t) - E_{p_\theta(a'_t|s_t)}[\phi(a'_t)] \end{aligned} \quad (4.3)$$

公式中价值函数 $R(h)$ 用来衡量得到的推导 (h) 的价值，价值越大，表明该推导越准确。 η 是学习率，我们利用 AdaGrad[46] 进行设置。

4.3.2 打分函数

为了选择最好的解析结果，我们设计了一个能够综合考虑句子重写部分和语义解析部分的打分函数。给定一个句子 x ，我们利用句子重写得到的新句子 x' ，利用语义解析器对 x' 进行解析，得到推导 (derivation) d ，我们的打分函数定义如下：

$$\begin{aligned} score(x, x', d) &= \theta \cdot \phi(x, x', d) \\ &= \theta_1 \cdot \phi(x, x') + \theta_2 \cdot \phi(x', d) \end{aligned}$$

打分函数分为两部分，一部分是句子重写部分—— $\theta_1 \cdot \phi(x, x')$ ；另一部分是语义解析部分—— $\theta_2 \cdot \phi(x', d)$ 。类似于 Berant and Liang [15]的方法，打分函数中语义解析部分的参数 θ_2 利用公式4.2进行更新。同理，打分函数中句子重写部分的参数 θ_1 利用如下公式进行更新：

$$\begin{aligned}\theta_1 &\leftarrow \theta_1 + \eta R(h_{target}^*) \delta(x, x'^*) \\ \delta(x, x'^*) &= \nabla \log p_{\theta_1}(x'^* | x) \\ &= \phi(x, x'^*) - E_{p_{\theta_1}(x'|x)}[\phi(x, x')]\end{aligned}$$

4.3.3 参数学习算法

为了学习参数 θ_1 和 θ_2 , 我们通过问题-答案对 (x_i, y_i) 来对模型进行训练。类似于 Berant and Liang [15], 我们通过最大化价值 (reward) 函数, 而不是最大化似然估计来学习参数。当然价值函数也提供了参数更新的方向。实际上, 每更新一次参数, 我们的模型就会使得到最高价值的推导 (derivation) 会有最高的打分, 简单来说, 就是价值函数与我们的打分函数是正相关的, 价值函数值越高, 表示打分函数值也高; 价值函数值越低, 表示打分函数值也低。表4.6显示了我们的训练算法。算法中函数 $REWRITING(x_i)$ 表示利用两种句子重写方法得到并返回新的句子集合; 函数 $PARSE(p_\theta, x)$ 表示利用当前的参数对句子进行解析; 函数 $CHOOSEORACLE(h_0)$ 表示从 h_0 的根节点中选择价值函数值最高的推导 (derivation); 函数 $CHOOSEORACLE(H_{target})$ 表示在候选集合中选择价值函数值最高的推导; 函数 $CHOOSEORACLE(h_{target}^*)$ 表示在拥有最高价值函数的推导中所使用的利用句子重写得到的新句子。

4.3.4 特征

上节已经提到, 我们有使用两类特征, 一类是语义解析部分的特征, 这里我们使用了基准语义解析器所使用的全部特征, 详细描述, 请参见 [15]。另一类是句子重写部分的特征, 这部分特征的定义是基于原句子, 新句子以及最后的解析 (derivations), 具体的定义如下:

基于字典替换的特征 给定一个句子 s_0 , 当我们用句子中某个词语的解释 ex 替换该词 w , 得到新的句子 s_1 的时候, 我们定义如下 4 个特征模板: 第一个特征是所替换的词语; 第二个特征是我们所使用的替换 $w \rightarrow ex$; 最后两个特征模板是被替换的词语 w 在原句子 s_0 中左边词语和右边词语的词性 (POS)。

基于模板替换的特征 给定一个句子 s_0 , 当我们使用复述模板对 $t_1 \rightarrow t_2$, 重写句子得到新句子 s_1 的时候, 我们也定义了如下 4 个特征模板: 第一个特征是我们所使用的模板对 (t_1, t_2) ; 第二个特征是句子 s_0 和模板 t_1 之间的相似度, 具体的我们使用 s_0 和 t_1 中词语的重叠程度来定义相似度; 第三个特征是复述模板对 (t_1, t_2) 的可信度, 具体的, 我们通过计算 WikiAnswers 语料中模板 t_1 和 t_2 的点互信息 (PMI); 第四个特征模板是目标语义表示中出现的二元谓词 p 和模板 t_2 之间的配对, 这个特征是为了捕捉模板 t_2 到谓词 p 的映射。

Input: Q/A 对 $\{(x_i, y_i) : i = 1 \dots n\}$; 知识库 \mathcal{K} ; 句子数目 N ; 迭代次数 T .

Definitions: 函数 $REWRITING(x_i)$ 利用句子重写方法由原句子 x 得到若干候选的新句子; 函数 $PARSE(p_\theta, x)$ 使用 agenda-parsing 方法基于当前的模型参数 θ 对句子 x 进行解析; 函数 $CHOOSEORACLE(h_0)$ 从根节点 h_0 处选择价值最高的推导; 函数 $CHOOSEORACLE(H_{target})$ 从若干候选推导中选择价值最高的推导; 函数 $CHOOSEORACLE(h_{target}^*)$ 选择得到最高价值推导的新句子。

Algorithm:

```

 $\theta_1 \leftarrow \mathbf{0}, \theta_2 \leftarrow \mathbf{0}$ 
for  $t = 1 \dots T, i = 1 \dots N$ :
     $X = REWRITING(x_i)$ 
    for each  $x'_i \in X$  :
         $h_0 \leftarrow PARSE(p_\theta, x'_i)$ 
         $d^* \leftarrow CHOOSEORACLE(h_0)$ 
         $h_{target} \leftarrow PARSE(p_\theta^{+cw}, x'_i)$ 
         $h_{target}^* \leftarrow CHOOSEORACLE(H_{target})$ 
         $x'^*_i \leftarrow CHOOSEORACLE(h_{target}^*)$ 
         $\theta_2 \leftarrow \theta_2 + \eta R(h_{target}^*) \sum_{t=1}^T \delta(h_{target}^*)$ 
         $\theta_1 \leftarrow \theta_1 + \eta R(h_{target}^*) \delta(x_i, x'^*_i)$ 

```

Output: Estimated parameters θ_1 and θ_2 .

表 4.6: 我们的参数更新算法

4.4 实验

这一节，我们介绍我们的实验，并于之前的方法进行对比，以及对实验结果进行分析。

4.4.1 实验设置

实验数据 我们在公开数据集 WEBQUESTIONS 上进行实验。该数据集覆盖多个领域，以 Freebase 为知识库本体。Yao [147]指出该数据集 85% 的样例都是简单问句：句子中包含一个关系词和一个实体，分别对应到知识库中的一个二元谓词和一个实体，问句的答案就是知识库中该二元谓词除了已知实体外的另外一个实体（也可以是多个）。我们使用标准的训练/测试数据分割。

另外，为了测试我们的方法在结构不匹配问题上的效果，我们在 WEBQUESTIONS 数据集上人工收集了 50 个存在结构不匹配的问题，如 “Who is keyshia cole dad?” 和 “What countries have german as the official language?”, 并在该数据集上进行了测试。

系统设置 实验中，我们利用 Freebase 的搜索 API 来辅助实体链接。我们把 Freebase 导入 Virtuoso³数据库，并将逻辑表达式转换为 SPARQL⁴查询语句，再在 Virtuoso 上进行查询，得到逻辑表达式所对应的答案。训练模型时，训练迭代次数我们设置为 3 次；解码的时候所使用的柱搜索的大小，我们设置为 200；句子重写部分，每一个句子利用基于词典替换的重写方法得到的新句子最多不超过 100 个，利用基于模板替换的句子重写方法得到的新句子同样最多不超过 100 个。

基准系统 我们与当时在该数据集上取得结果最好的几个系统进行了对比：这些系统包括基于语义解析的系统 [16, 14, 15, 149]；基于信息抽取的系统 [148, 147]；基于机器翻译的问答系统 [11]；基于词向量表示的问答系统 [19, 146]。所有这些系统的结果都取自他们的论文或者标准网站⁵。

评估 (evaluation) 与 Berant et al. [16]的评估标准一样，我们使用问题答案的准确率作为评估标准。由于标准答案中可能不止一个实体，会有多个实体的情况，因此我们采用平均 F1 作为我们的主要评估标准。该评估标准也是公认的，所有系统都采用这个评估标准。

4.4.2 实验结果与分析

表4.7展示了我们的方法和之前的方法的结果对比，从表中我们可以看到：

³<https://www.virtuoso.com/>

⁴<https://www.w3.org/TR/rdf-sparql-query/>

⁵<https://worksheets.codalab.org/worksheets/0xba659fe363cb46e7a505c5b6a774dc8a/>

System	Prec.	Rec.	F1 (avg)
Berant et al. [16]	48.0	41.3	35.7
Yao and Van Durme [148]	51.7	45.8	33.0
Berant and Liang [14]	40.5	46.6	39.9
Bao et al. [11]	—	—	37.5
Bordes et al. [19]	—	—	39.2
Yang et al. [146]	—	—	41.3
Bast and Haussmann [12]	49.8	60.4	49.4
Yao [147]	52.6	54.5	44.3
Berant and Liang [15]	50.5	55.7	49.7
Yih et al. [149]	52.8	60.7	52.5
我们的方法	53.7	60.0	53.1

表 4.7: 我们的方法与之前的方法在 WEBQUESTIONS 数据集上的实验结果

1. 我们方法的结果相当好：我们的方法超过了当时所有的系统，在 WEBQUESTIONS 数据集上取得了最高的 F1 值 53.1。
2. 句子重写技术能有效提高语义解析的性能：通过利用句子重写，我们的方法比基准语义解析器 (Berant et al. [16]) 在 F1 值上提高了 3.4 个点 (53.1 VS 49.7)。
3. 与所有系统对比，我们的方法取得了最高的准确率 (P 值)。这个结果也表明利用句子重写，我们能够得到更加准确的逻辑表达式。另外，我们方法在召回率上也表现十分出色，排在当时所有系统中的第三名，而两个排名在我们之前的系统都有使用额外的技术 (实体链接和关系匹配)。

在结构不匹配问题上的效果分析 为了分析结构不匹配问题的普遍性，我们在 WEBQUESTIONS 数据集的训练集上随机抽取了 500 个句子，并进行人工分析。我们发现样本数据中总用有 12.2% 的句子存在结构不匹配问题：其中 3.8% 的句子属于 1-N 型不匹配问题；8.4% 的句子属于 N-1 型不匹配问题。

为了进一步验证我们方法对于结构不匹配问题的有效性，我们在 50 个存在结果不匹配问题的小测试集上进行了实验，表 4.8 展示了在该小测试集上的测试结果。从结果中我们可以看到，我们的方法能够十分有效的解决结构不匹配问题：先比基准语义解析器，我们基于句子重写的语义解析器在 F1 值上，整整提高了 54.5 个百分点。

两种句子重写方法的效果分析 为了对比两种句子重写方法的效果，我们进行如下对比实验设置：不使用句子重写；仅使用基于字典替换的句子重写；仅使用基于

System	Prec.	Rec.	F1 (avg)
Base system	31.4	43.9	29.4
Our system	83.3	92.3	83.9

表 4.8: 在 50 个存在结构不匹配问题的测试集上的测试结果对比

Method	Prec.	Rec.	F1 (avg)
不使用句子重写	49.8	55.3	49.1
仅使用基于字典替换的句子重写	51.6	57.5	50.9
仅使用基于模板替换的句子重写	52.9	59.0	52.3
同时使用两种句子重写	53.7	60.0	53.1

表 4.9: 对比两种句子重写方法的效果

模板替换的句子重写；同时使用两种句子重写。实验结果如表4.9所示，从结果中我们可以看到：

1. 两种句子重写方法都能提高语义解析的性能，相比于基准系统（不使用句子重写方法），利用基于字典替换的句子重写的语义解析器和基于模板替换的句子重写的语义解析器在 F1 值上能够分别提高 1.8 和 3.2 个点。
2. 相比于基于字典替换的句子重写方法，基于模板替换的句子重写方法对语义解析的提高更多。我们认为主要是因为 N-1 型不匹配问题更加常见（在 WEBQUESTIONS 数据集上）。
3. 两种句子重写方法能够兼容并相互补充，同时使用两种句子重写的语义解析器取得了最好的结果。

我们还对句子重写方法的效果还进行了更进一步的分析，我们发现句子重写方法能够带来一些额外的收益，接下来我们将详细介绍，同时表4.10展示了几个例子，表格中 **O** 表示原来的句子；**R** 表示利用句子重写得到的新句子（模型选择出来的最好的）；**LF** 表示逻辑表达式。

提高系统鲁棒性的效果分析 我们发现：基于模板替换的句子重写方法能够有效提高基准语义解析器的鲁棒性。具体地，基于模板替换的句子重写方法会把相似的句子改写成统一模板，同时我们还定义了（模板，谓词）的特征，这样一来，能够消除解析过程中的一些不确定性。例如，对于句子：“*What does jamaican people speak?*”，其准确的逻辑表达式为：`language_spoken.jamaican`。若仅仅利用词语“*people*”和“*speak*”到谓词 `official_language` 和 `language_spoken` 之间的不确定映射关系，基准语义解析器会把句子映射到 `official_language.jamaican`

O	<i>Who is willow smith mom name?</i>
R	<i>Who is willow smith female parent name?</i>
LF	<code>parentOf.willow_smith^gender.female</code>
O	<i>Who was king henry viii son?</i>
R	<i>Who was king henry viii male child?</i>
LF	<code>childOf.king_henry^gender.male</code>
O	<i>What are some of the traditions of islam?</i>
R	<i>What is of the religion of islam?</i>
LF	<code>religionOf.islam</code>
O	<i>What does jamaican people speak?</i>
R	<i>What language does jamaican people speak?</i>
LF	<code>language_spoken.jamaica</code>

表 4.10: 若干基于句子重写的语义解析器比基准语义解析器表现更好的例子

(详细的例子请查看表4.10中最后一个例子)。而我们的系统，通过利用模板“*what language does \$y people speak*”与谓词 `language_spoken` 之间的映射关系(定义的特征)，我们的系统能够准确的解析这种例子。

解决未登录词 (OOV) 的效果分析 我们发现利用句子重写的方法能够在某种程度上为解决未登录词 (OOV) 问题提供一点帮助。传统的方法中，如果句子中的词语没有被词典 (lexicon) 覆盖掉，那么这个例子很难被解析正确。但是，通过句子重写，我们可能会把未登录词替换了被词典覆盖的词，从而可以利用词典对重写的句子进行解析。例如，表4.10中的第 3 个例子，句子 “*What are some of the traditions of islam?*” 中有未登录词 “*tradition*”，从而不能被基准语义解析器正确的解析。而我们利用句子重写的方法，得到新的句子 “*What is of the religion of islam?*”，新句子中的核心内容词 “*religion*” 是被词典所覆盖的，从而这个例子可以被我们的方法正确解析。

4.4.3 错误分析

为了进一步了解我们的系统，我们对我们系统的结果进行了错误分析。具体地，我们从错误样例中随机选择了 100 个样例进行统计和分析。我们发现我们系统的错误的原因主要有如下这 4 种 (表4.11):

第一大原因是标注问题 (label issue)。主要的标注问题是标注不完全，就是说标注的答案不完全，只是准确答案的一部分。如，对于问句 “*Who does nolan ryan play for?*”，我们系统给出的答案是 4 个队 (正确的也是这 4 个队)，而标注的答案

Reason	#(Ratio)	Sample Example
Label issue	38	<i>What band was george clinton in?</i>
N-ary predicate($n > 2$)	31	<i>What year did the seahawks win the superbowl?</i>
Temporal clause	15	<i>Who was the leader of the us during wwii?</i>
Superlative	8	<i>Who was the first governor of colonial south carolina?</i>
Others	8	<i>What is arkansas state capitol?</i>

表 4.11: 系统出现错误的主要原因以及出现的比例和样例

只有 2 个队。另外的标注问题是错误，比如，对于问句 “*What state is barack obama from?*”，标注的答案是 “*Illinois*”，而准确的答案是 “*Hawaii*”。

第二大原因是多元谓词 (n-ary predicate, $n > 2$) 问题。目前，还很难让一个语义解析器准确生成一个包含多元谓词的逻辑表达式，如问句 “*What year did the seahawks win the superbowl?*” 描述的是一个包含多个元素的有关冠军的事件，给出这个冠军事件的当事者，和这个冠军的名目，寻求这个冠军事件发生的年份。这是一个比较复杂的样例，但是现实中很多提出的问题都是关于事件的，而知识库本体中对于事件的表示就是使用多元谓词，所以我们呼吁学者把注意力放到这类复杂问题上。

第三大原因是时间从句 (temporal clause)。就是问题中出现了时间从句，比如问句 “*Who did nasri play for before arsenal?*” 中的 “*before arsenal*”。这类问题对于当前的语义解析器是很难处理的，主要是因为没有一个特别合适的机制来建模和处理这类问题，这也是以后需要注意的地方。

第四大原因是最高级 (superlative) 问题，这对于语义解析来说，同样是很困难的问题。比如问句 “*What was the name of henry viii first wife?*”，我们需要先符合要求的列表中按照时间顺序选择第一个。这对于语义解析器来说十分困难，一是要弄清楚排序的对象是什么，二是要弄清楚排序的标准是什么，是根据什么来排序的。

还有一些原因是很难总结的，比如拼写错误，语料中拼写的例子有把 “*capital*” 拼成 “*capitol*”，把 “*marry*” 拼成 “*mary*”。

4.5 本章小结

本章从结构预测的角度研究语义解析。针对语义解析中结构预测部分存在的自然语言句子与目标语义表示之间结构不匹配的问题，我们提出了一种基于句子重写的语义解析方法。该方法的出发点是：一个句子总存在一种表达，该表达保留了原句子的意思，并且与目标语义表示之间的结构是匹配的。我们对结构不匹配问题进行了归纳并分类为：1-N 型和 N-1 型。针对 1-N 型不匹配问题，我们提出了

一种基于字典替换的句子重写方法；针对 N-1 型不匹配问题，我们提出了一种基于模板替换的句子重写方法。我们将句子重写方法应用于语义解析中。在公开数据集上进行了实验，实验结果验证了我们方法的有效性：句子重写的方法能够解决语义解析中句子与目标语义表示之间结构不匹配的问题。

本章的相关研究成果“**Sentence Rewriting for Semantic Parsing**”以长文的形式发表于 **ACL-2016 (CCF-A 类)**。

第五章 基于端到端语义图生成的语义解析

5.1 本章引论

前面已经介绍，一个完整的语义解析器一般需要两个功能：结构预测和语义落地。传统的语义解析器大部分都基于组合文法，如组合范畴文法（CCG）[154, 152]、基于依存的组合语义文法（DCS）[93]。这些语义解析器都利用文法来组合结构，利用词典来进行语义落地，再利用特征对候选解析结果打分排序。然而，在面向开放域的情况下，一般很难设计文法，更难学习到高质量、高覆盖度的词典，此外，要想设计出有效的特征通常也很困难，并且模型的学习也不是端到端的。为了解决上述问题，近年来语义解析领域内有两个备受关注的技术发展路线：基于语义图的方法，和基于序列-序列的方法。

基于语义图的方法 [123, 124, 12, 149] 利用语义图来表示自然语言句子的语义（例如知识库中的一个子图，如图5.1所示），进而把语义解析转化为语义图匹配/生成的问题。相比于逻辑表达式，语义图与知识库更加紧密相连 [149]，并且语义图的结构与句子的句法结构具有很多的共同点 [123]。同时，来自知识库的结构约束和语义约束能够很好的在解析的过程中加以利用 [149]。基于语义图的方法的最大挑战在于如何去构建一个句子的语义图。现阶段，语义图要么是利用特定的模板，再加上模板匹配而来 [12]，要么从依存树或者句法树转化而来 [123, 124]，或者通过启发式的算法 [149]。这些方法都是基于人工设计的，或者启发式的构建过程，这使得这些方法很难处理开放域条件下的复杂句子。

近年来，循环神经网络（RNN）模型在序列-序列的问题上已经取得了很大的成功，如机器翻译 [7]、AMR 分析 [76]，这得益于这类模型拥有强大的表示能力和预测能力。也有不少工作把序列-序列的模型应用到语义解析任务上来 [144, 43, 68, 61]，这些工作利用循环神经网络模型把句子被解析成序列化的逻辑表达式。这样一来不再需要特定的文法，也不需要学习高质量的词典，更不需要定义有效的特征，并且模型的学习是端到端的，此外，注意力机制 [7, 99] 可以用来学习句子中的词语与逻辑表达式中的谓词之间的软（soft）对齐。

在本章，我们提出了一种新的语义解析框架——Sequence-to-Action，该框架能够同时利用到语义图的强语义表示能力，以及循环神经网络模型的强表示学习能力和序列预测能力。具体地，我们把语义解析问题建模成一个端到端的语义图生成的过程。如图5.1中的例子所示，对于给定的句子 “*Which states border Texas*”，我们的模型会把它解析成一个动作序列：[add_variable:A, add_type:state, ...]。为了达到上述目标，首先我们需要设计一个动作集合，包括添加节点动作，比如：添加-变量节点（add_variable）、添加实体节点（add_entity）和添加类别节点

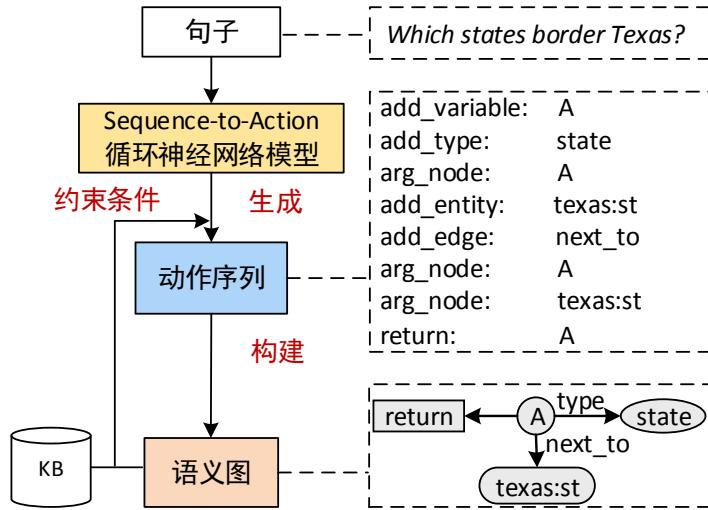


图 5.1: 我们方法的概览，其右侧是一个示例

(`add_type`)，和添加边动作，如：添加-边 (`add_edge`)，还有添加操作符动作，如：添加-最大操作符 (`add_argmax`)、添加-最小操作符 (`add_argmin`)、添加-计数操作符 (`add_count`) 等，该动作集合能够编码语义图的构建。然后，我们设计一个循环神经网络模型来生成动作序列，该动作序列用于构建句子所对应的语义图。最后，我们还探究了在解码的过程中加入句法约束条件和语义约束条件，用来及时过滤解码过程中的错误动作，进一步提高解析的准确度。

之前基于语义图的方法都使用人工设定的、或者启发式的生成算法来构建语义图，相比他们的方法，我们的 sequence-to-action 模型能够利用循环神经网络模型来生成动作序列，而动作序列可以用来构建语义图，模型的训练是端到端的 (end-to-end)。这样一个可以进行端到端学习的生成形式使得我们的方法更有有效，同时能够适应到更多的不同的情形。

而相比如之前的基于序列-序列 (Seq2Seq) 的语义解析方法，我们的方法使用循环神经网络模型来生成动作序列，而不是序列化的逻辑表达式。我们发现，我们所采用的动作编码形式能够捕捉到更多的句法信息和语义信息，同时我们的编码方式更加的紧凑。此外，我们可以加入句法约束条件和语义约束条件来进一步增强语义解析。例如，在 GEO 数据集中，动作 `add_edge:next_to` (关系：接壤) 必须满足如下语义约束条件：这条边所对应的两个节点的类别只能分别是 `state` (类别：州) 和 `state`，还必须满足如下句法约束条件：边 `next_to` 必须连接两个节点，以构成一个有效的图。

我们在三个公开数据集上进行了对比实验，这三个数据集分别为：GEO [151]，ATIS [60] 和 OVERNIGHT [139]。实验结果论证了我们方法的有效性：我们的方法在 ATIS 和 OVERNIGHT 这两个数据集上都取得了现阶段最好的结果，在 GEO 数据集上也取得了很有竞争力的结果。

本章工作的主要贡献可以总结为以下两点：

1. 我们提出了一个新的语义解析框架——Sequence-to-Action，该框架把语义解析问题建模成端到端的语义图生成过程。这个新的框架能够同时利用语义图的语义表示能力和循环神经网络模型的强序列预测能力。
2. 我们设计了一个 sequence-to-action 的生成模型，和一个动作集合用来编码语义图的构建，还有一个循环神经网络模型用来生成动作序列。此外，我们还利用句法和语义条件约束来进一步增强语义解析。实验结果也论证了我们方法的有效性。

5.2 端到端的语义图生成方法

给定一个句子 $X = x_1, \dots, x_{|X|}$ ，我们的 sequence-to-action 模型将生成一个动作序列 $Y = y_1, \dots, y_{|Y|}$ ，该动作序列将用于构建句子所对应的语义图。图5.2展示了一个例子。我们的模型中，由 X 生成 Y 的条件概率 $P(Y|X)$ 利用链式法则可以分解为如下形式：

$$P(Y|X) = \prod_{t=1}^{|Y|} P(y_t|y_{<t}, X) \quad (5.1)$$

其中 $y_{<t} = y_1, \dots, y_{t-1}$ 。

为了达到上述目标，我们需要：1) 一个动作集合，该集合能够编码语义图的构建；2) 一个编码器（encoder），该编码器把自然语言句子 X 转化为一个向量表示，和一个解码器（decoder），该解码器根据编码器传过来的向量表示来生成动作序列 $y_1, \dots, y_{|Y|}$ 。接着，我们将分别介绍这两部分。

5.2.1 用于语义图构建的动作集

通常，一个语义图有如下几个成分组成：节点（包括变量节点，实体节点和类别节点）、边（对应到语义关系）和一些通用的操作符（如，最大操作符、最小操作符、计数操作符、求和操作符等）。为了构建一个语义图，我们设计了六种类型的动作，它们分别是：

添加-变量节点（Add Variable Node:） 这类动作表示往语义图上添加一个变量节点。一般情况下，变量节点同时也是返回的节点（对应到疑问词：*which*, *what*），但也有可能是一个中间变量节点。我们使用如下形式来表示这类动作：`add_variable:A`，其中 `A` 是变量节点的标识 ID。

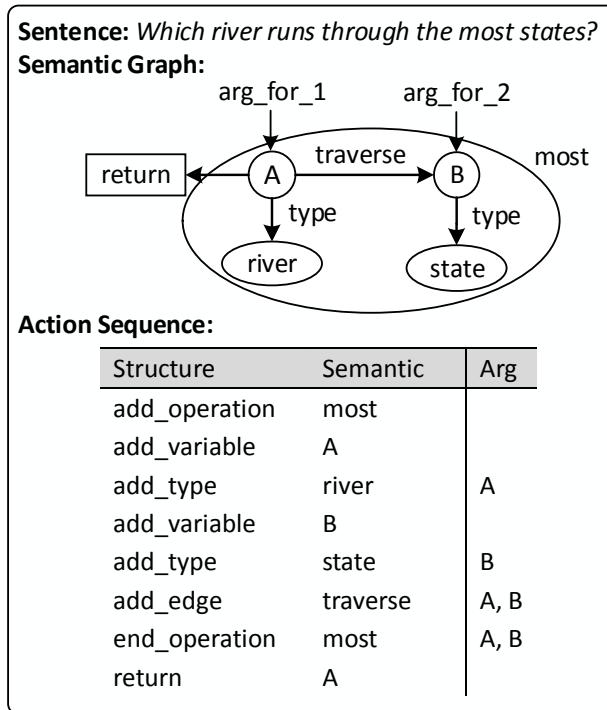


图 5.2: 例子展示: 一个句子, 以及该句子所对应的语义图和用来构建该语义图的动作序列

添加-实体节点 (Add Entity Node:) 这类动作表示往语义图上添加一个实体节点。一个实体节点对应到知识库中的一个实体。我们使用如下形式来表示这类动作: `add_entity_node:texas`。

添加-类别节点 (Add Type Node:) 这类动作表示往语义图上添加一个类别节点, 如(*state, city*)。我们使用如下形式来表示这类动作: `add_type_node:state`。实际上, 类别节点本来与变量节点和实体节点是不同的, 一个类别节点其实是包括一个表示类别的节点, 和一条边, 只是这条边的值都是一样的, 都是类别 `Type`, 为了简化一个类别节点, 我们才设计成上面的样子。

添加-边 (Add Edge:) 这类动作表示往语义图上的两个节点之间添加一条边。一条边对应到知识库中的一个二元关系。我们使用如下形式来表示这类动作: `add_edge:next_to`。

操作符动作 (Operation Action:) 这类动作表示往语义图上加一个操作符。一个操作符可以是: 最大 (`argmax`)、最小 (`argmin`)、计数 (`count`)、求和 (`sum`)、否定 (`not`) 等 (复杂的情况还有比较 (`compare`), 这里我们先不考虑)。由于每一个操作符都有它的辖域, 为了表示一个操作符的辖域, 我们定义为每一个操作符定义两个动作: 一个开始操作符, 表示如: `start_operation:most`, 一个结束操作符, 表示如: `end_operation:most`。这两个操作符之间的动作序列所构成的语义子图就是该操作符的辖域。在显示的语义图上, 我们用一个圈来表示该操作符的辖域。

参数动作 (Argument Action:) 由于上述的动作中有些类别的动作需要额外

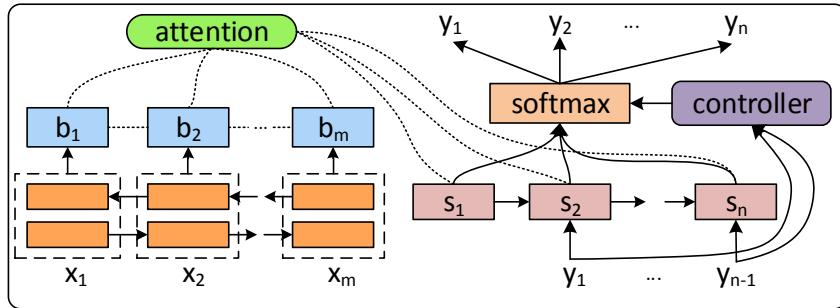


图 5.3: 基于注意力机制的用于生成动作序列的循环神经网络模型, 附带一个添加约束条件的控制器

的参数信息, 如, 一条边 `add_edge:next_to` 是连接的哪两个节点。因此, 我们分别为添加-类别节点 (`add_type`)、添加边 (`add_edge`) 和操作符 (`operation`) 设计了参数动作, 并且参数动作总是跟在它的主动作的后面。

对于添加-类别节点 (`add_type`) 动作, 我们添加一个参数动作, 这个参数动作用来表明该类别是用来约束哪个节点的这个参数可以是变量节点, 可以是实体节点。这类动作可以表示为: `arg:A`。

对于添加边(`add_edge`)动作, 我们添加两个参数动作:`arg1_node` 和 `arg2_node`, 用来表示该边连接哪两个节点。这类动作可以表示为:`arg1_node:A` 和 `arg2_node:B`。

对于不同的操作符 (`operation`), 我们又设计了不同的参数, 对于求和操作 (`operation:sum`), 我们添加 3 个参数动作: `arg-for`、`arg-in` 和 `arg-return`, 分别用来表明求和操作所作用的节点, 所统计的因素和返回的结果 (一般为一个变量节点); 对于计数操作 (`operation:count`), 我们添加两个参数动作: `arg-for` 和 `arg-return`, 分别用来表明所要计数的节点和返回的结果; 对于最高级操作 (`operation:most`), 我们添加 2 个参数动作: 均为 `arg-for`, 用来表明所作用的两个节点之间的边, 并且这两个节点有先后顺序。

我们可以看到每一个动作都同时建模了句法和语义信息, 这使得模型在使用这种编码形式的时候能够捕捉到更多的信息, 同时这些信息与知识库保持着紧密的联系。另外, 我们还发现使用动作序列的编码形式比直接序列化逻辑表达式的形式更加的紧凑 (详情参加5.4.3节)。

5.2.2 词语序列-动作序列的神经网络模型

基于上节介绍的动作编码, 这一节我们介绍我们用于把句子解析成动作序列的编码-解码 (encoder-decoder) 模型。具体地, 类似于 Jia and Liang [68]所使用的循环神经网络模型, 我们这里也利用基于注意力机制的序列-序列的循环神经网络模型。图5.3展示了该模型的示例图。

编码器 (encoder): 编码器负责使用双向循环神经网络 [7] 把输入的句子 x_1, \dots, x_m

转化为包含上下文信息的向量表示序列 b_1, \dots, b_m 。首先，每一个词 x_i 被映射成它的向量化表示，然后这些向量表示被输入到两个循环神经网络：一个前向循环神经网络（forward RNN）和一个后向循环神经网络（backward RNN）。最后我们递归使用如下计算方式来生成隐藏状态的序列 h_1, \dots, h_m ：

$$h_i = LSTM(\phi^{(x)}(x_i), h_{i-1}) \quad (5.2)$$

其中我们采用长短记忆时记忆网络（LSTM）[62] 的形式来进行计算。最后，对于输入句子的每个位置 i ，我们定义它的带有上下文信息的向量化表示为： $b_i = [h_i^F, h_i^B]$ 。

解码器（decoder）：我们使用经典的基于注意力机制的解码器 [7]，该编译器依次生成一个动作，构成动作序列 y_1, \dots, y_n 。在时间节点 j ，编译器基于当前的隐藏状态 s_j 生成动作 y_j ，接着基于 s_j 和 y_j 更新隐藏状态，得到 s_{j+1} 。这个编译器是由如下计算公式形式化定义的：

$$s_1 = \tanh(W^{(s)}[h_m^F, h_1^B]) \quad (5.3)$$

$$e_{ji} = s_j^T W^{(a)} b_i \quad (5.4)$$

$$a_{ji} = \frac{\exp(e_{ji})}{\sum_{i'=1}^m \exp(e_{ji'})} \quad (5.5)$$

$$c_j = \sum_{i=1}^m a_{ji} b_i \quad (5.6)$$

$$P(y_j = w | x, y_{1:j-1}) \propto \exp(U_w[s_j, c_j]) \quad (5.7)$$

$$s_{j+1} = LSTM([\phi^{(y)}(y_j), c_j], s_j) \quad (5.8)$$

其中归一化后的注意力分数 a_{ji} 定义了在输入的词上的概率分布，词上面的每一个概率值表示在时刻 j 在该词上的注意力集中程度，概率值越大，表明注意力越集中在该词上面。另外， e_{ji} 是未归一化的注意力分数。

为了能够在解码的过程中加入约束条件，我们在模型中加入了一个控制器（controller），我们将在第5.3.3节详细介绍该控制器。

动作的向量表示：在上述解码器中需要每一个动作的向量化表示 (y_j)。如前面介绍的，每一个动作都包含两部分：一部分是句法部分（如，`add_edge`），另一部分是语义部分（如，`next_to`）。这样一来，动作与动作可能会共享句法部分或者语义部分。如动作 `add_edge:next_to` 和动作 `add_edge:loc` 就共享了句法部分，而动作 `add_node:A` 和动作 `arg_node:A` 就共享了句法部分。为了减少参数的数量，尽量是参数不稀疏，我们先让每个句法部分和每个语义部分都单独具有它的向量化表示，对于一个动作，我们把它的句法部分和语义部分的向量串联起来（concatenate），组成该动作的向量化表示，比如 $\phi^{(y)}(\text{add_edge:next_to}) = [\phi_{strut}^{(y)}(\text{add_edge}), \phi_{sem}^{(y)}(\text{next_to})]$ 。其中动作的向量映射函数 $\phi^{(y)}$ 通过训练学习而来。



图 5.4: 逻辑表达式与动作序列之间相互转换的流程简介图

5.3 带约束条件的神经语义解析模型

在这一节，我们将描述我们是如何利用 sequence-to-action 模型来建立一个语义解析器的。我们首先介绍我们如何训练我们的模型，以及如何进行推理（inference），接着介绍如何在解码的过程中加入句法约束条件和语义约束条件。

5.3.1 训练

参数学习 我们模型的参数包括：循环神经网络模型的参数 $W^{(s)}$ 、 $W^{(a)}$ 和 U_w ，词向量映射函数 $\phi^{(x)}$ ，和动作向量映射函数 $\phi^{(y)}$ 。我们通过训练语料来学习这些参数。给定一个训练样例（句子 X 和它所对应的动作序列 Y ），我们最大化 X 由生成 Y 的似然估计。我们的目标函数如下（对其最大化）：

$$\sum_{i=1}^n \log P(Y_i|X_i) \quad (5.9)$$

我们使用标准的随机梯度下降算法来更新参数。

逻辑表达式到动作序列的转换 现阶段用于语义解析的语料大多都是以逻辑表达式来标注的。为了训练我们的模型，我们需要得到句子的语义图所对应的动作序列。我们以语义图作为中间表示把逻辑表达式转换为动作序列（图5.4展示了转换的一个简单流程图）。具体地，首先我们利用深度优先的算法把逻辑表达式转换成语义图，再使用相同的顺序从语义图生成动作序列。其中实体，变量和类别都是节点，关系是边。其中作为中间表示的语义图，我们是使用专门的数据结构来进行表示的，类似于函数-参数的形式，边是函数名，节点是函数的参数，语义图上的一条边和两个节点对应到一组函数-参数的存储结构，整个语义图就对应到函数-参数的存储结构的一个序列，并且是要考虑该序列的顺序的。一方面，我们可能需要句子利用句子的语义表示在知识库上执行计算，得到信息来辅助训练，另一方面由于语义图的形式，和动作序列的形式都不容易直接理解，为了方便分析，我们也需要句子所对应的逻辑表达式，我们使用相同的策略，再从动作序列转换到逻辑表达式。

对于实体的处理机制 实体在语义解析中充当了十分重要的角色 [149]，之前的很多工作都直接使用手工实体词汇，当然也有使用实体链接技术来实现句子中实体的语义落地的 [83]。在 Dong and Lapata [43]的工作中，他们使用预处理的方式，先利用字符串匹配的方法把实体都识别出来，由于实体容易带来数据稀疏的问题

(一般一个实体在语料中出现的次数会很少), 他们用实体的类别和独有 ID 来表示实体。如 *Texas* 会表示为 *State0*。在 Jia and Liang [68]的工作中, 他们使用神经机器翻译里面经常使用的复制 (Copying) 机制来处理实体 (他们还需要一个实体词典)。在本章的文章, 我们都尝试并实现两种实体处理的机制, 并在后面的实验中比较了这两种实体处理机制。

5.3.2 推导

在测试的时候, 给一个新的句子 X , 我们通过如下公式来得到它所对应的动作序列 Y :

$$Y^* = \operatorname{argmax}_Y P(Y|X) \quad (5.10)$$

其中 $P(Y|X)$ 的计算是利用(5.1)计算而来。在解码的时候, 我们使用了柱搜索 (beam search)。我们可以从最优动作序列 Y^* 得到相应的语义图和逻辑表达式。

5.3.3 在解码中加入约束条件

在解码的时候, 我们依次生成一个动作。显然, 动作与动作之间是有很强的关联的, 也就是说下一个动作与当前生成的语义子图是有关联的, 有一些动作明显是错误的, 可以利用句法约束条件和语义约束条件来进行刷选。已经有工作证明了句法和语义约束条件可以用来增强语义解析。Krishnamurthy et al. [83]和 Iyyer et al. [67]都利用类别约束条件来过滤错误的符号 (token); Liang et al. [90]利用一个 Lisp 解释器和预定义的函数来生成合法的符号 (token)。在本章我们也利用句法和语义约束条件, 具体的, 我们设计了一个控制器, 该控制器读入已经生成的动作序列, 并构建语义子图, 然后利用如下的句法约束条件和语义约束条件来对动作进行过滤, 把违背约束条件的动作滤掉。

句法约束条件 句法约束条件是为了保证生成的动作序列可以构成一个连通的无环图, 这是因为语义图一般是连通的, 也是无环的。比如, 没生成一个添加边的动作, 该边都必须有两个不同的节点作为它的参数, 图5.5中, 作为下个动作的候选动作中第三个动作就是违背了这条约束规则, 它的两个参数节点是一样的, 会导致生成的图有环, 不符合语义图的要求。这种类型的约束条件是领域无关的, 并且是通用的。我们在控制器中通过若干条规则来建模句法约束条件。

语义约束条件 语义约束条件是为了保证生成的语义图符合知识库本体模板框架 (schema) 的约束。具体的, 我们建模了两类语义约束条件, 一类是选择约束 (selectional preference), 这种约束条件是确保生成的语义图中边的两个参数 (节点) 的类型必须符合知识库中本体模板框架 (schema) 中类型的约束。比如, 在 GEO 数据集中, 边 `next_to` (关系: 接壤) 的两个节点参数的类型均是 `state`

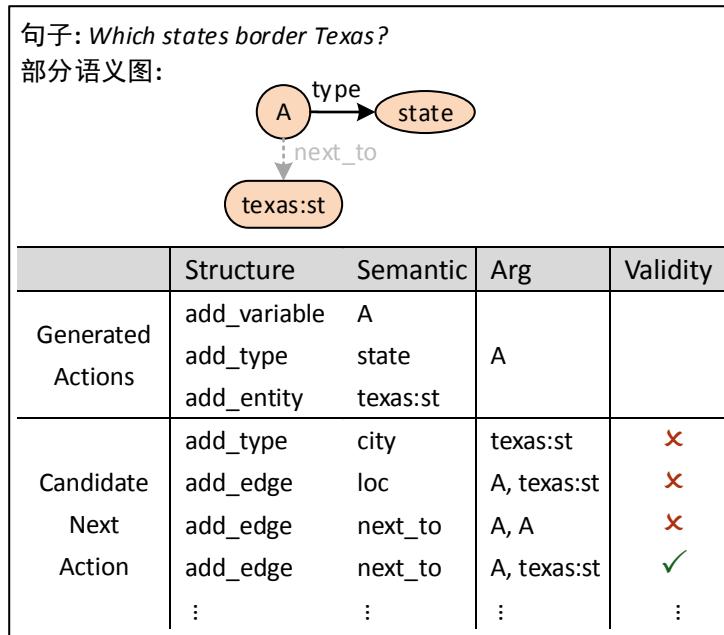


图 5.5: 使用约束条件来过滤错误动作的示例

(州)。图5.5中，作为下个动作的候选动作中第二个动作违背了这条约束规则，由于边 `loc` (关系: 位于) 的两个参数节点类型可以分别是 `city` (类别: 城市) 和 `state` (类别: 州)，而例子中，第一个参数节点 (变量节点 `A`) 的类型是 `state` (类别: 州)，而第二个参数节点 (实体节点 `texas:st`) 的类型也是 `state` (类别: 州)，这就违背了选择约束。第二类约束是类别不冲突约束条件，这种约束条件保证每个节点 (可以是变量节点，也可以是实体节点) 的类别不冲突，比如说一个节点的类型不能同时是 `city` (类别: 城市) 和 `state` (类别: 州)。图5.5中，作为下个动作的候选动作中第一个动作违背了这条约束规则，由于这个动作是在实体节点 `texas:st` 加一个类别节点 (`city`)，而实体节点 `texas:st` 的类型已经是 `state` (类别: 州) 了，这就违背了节点的类型不冲突约束条件。语义约束条件是领域相关的，不过我们可以从知识库中自动抽取出这类约束条件。在控制器中，我们同样使用若干条规则来编码语义约束条件。

5.4 实验

这一节，我们将在公开数据集上进行实验，并与之前的方法进行对比，来验证我们的方法是否有效。我们在三个公开数据集上进行了实验: GEO、ATIS 和 OVERNIGHT。

5.4.1 实验设置

我们所使用的循环神经网络模型中，隐藏状态维度为 200，输入句子中的词的向量维度是 100。前面已经介绍过，每一个动作的向量表示是该动作中句法部分的向量表示和语义部分的向量表示的串联。而句法部分的向量维度和语义部分的向量维度我们是通过统计语料中动作序列的句法部分与语义部分的个数比例来进行预设置，具体的，两者的维度总和也是 100 维左右，而维度的比例与它们在语料中个数的比例是一样的，最后我们在开发集上进行尝试，选择最好的向量维度设置。最终，在 GEO 数据集上，句法部分的维度是 20，语义部分的维度是 80；在 ATIS 数据集上，句法部分的维度是 10，与一部分的维度是 90；在 OVERNIGHT 数据集上，则在八个领域分别有对应的设置。所有参数都使用初始化的形式，并没有利用预训练的词向量等，具体的每个参数都是在区间 $[-0.1, 0.1]$ 上进行均匀采样得来。我们总共迭代训练 30 次，初始学习率设置为 0.1，然后第一组迭代 15 次，学习率衰减一半，后面每迭代 5 次，学习率依次衰减一半。对于语料中词语出现次数只有一次的我们用特殊符号 ($<\text{unk}>$) 进行代替，该特殊符号也有它的初始化向量表示。在解码的时候，我们使用了柱搜索 (beam search)，柱搜索的大小我们设置为 5。我们的模型是基于 Theano[17] 实现的。

评测的时候，我们所采用的评测指标和策略都是基于标准的指标和策略，如 Jia and Liang [68]一样，我们在 GEO 数据集和 OVERNIGHT 数据集上，我们在候选的动作序列中，我们选择能够在知识库中查询得到答案的排名最靠前的动作序列作为我们的预测结果。这里我们就会用到从动作序列转换到逻辑表达式，再利用逻辑表达式在知识库中进行查询。

5.4.2 实验结果与分析

我们在三个数据集上与当前最好的一些系统进行对比，因为所有的系统在三个数据集上的训练/测试语料分割都采用的同样的标准，所以所有的结果都是可比的。此外，有些系统在一个数据集上可能有不同的实验设置，我们只选择他们最好的结果，所有系统的结果都来自他们发表的论文。

对于我们的模型，我们有三种不同的设置，一种是最基本的 sequence-to-action 模型，没有使用任何约束条件的——Seq2Act；第二种是加上语法约束条件的——Seq2Act (+C1)；第三种是同时加上语法约束条件和语义约束条件的 Seq2Act (+C1+C2)。我们没有设置单独加语义约束条件的，这是因为我们所使用的语义约束条件是在符合句法约束条件的基础上添加的，如果一个语义图不符合句法约束条件，再来谈语义约束条件是没有意义的。在三个数据机上的结果分别在表5.1和表5.2中展示，两个表格中带 * 号的系统表示该系统有使用额外的语料或资源。从结果我们可以看到：

	GEO	ATIS
Previous Work		
Zettlemoyer and Collins [154]	79.3	—
Zettlemoyer and Collins [152]	86.1	84.6
Kwiatkowski et al. [85]	88.9	—
Kwiatkowski et al. [86]	88.6	82.8
Liang et al. [93]* (+lexicon)	91.1	—
Poon [118]	—	83.5
Zhao et al. [157]	88.9	84.2
Seq2Seq Models		
Jia and Liang [68]	85.0	76.3
Jia and Liang [68]* (+data)	89.3	83.3
Dong and Lapata [43]: 2Seq	84.6	84.2
Dong and Lapata [43]: 2Tree	87.1	84.6
Our Models		
Seq2Act	87.5	84.6
Seq2Act (+C1)	88.2	85.0
Seq2Act (+C1+C2)	88.9	85.5

表 5.1: GEO 数据集和 ATIS 数据集上的测试结果

1) 通过同时利用语义图的语义表示能力和循环神经网络模型的表示学习能力以及对序列问题的强预测能力，我们的方法（Seq2Act (+C1+C2)）在 ATIS 数据集和 OVERNIGHT 数据集上都取得了当前最好的结果，同时在 GEO 数据集上也取得了与最好成绩不相上下的结果。具体的，在 GEO 数据集上，我们完整的模型 Seq2Act (+C1+C2) 在相同的设置条件下取得的结果（88.9 的准确率）也是最好的，只落后于两个系统，一个是 Liang et al. [93]*，该系统有使用额外的人工定义的词汇；另一个是 Jia and Liang [68]*，该系统使用数据重组（data recombination）的方式对训练语料进行了扩充。在 ATIS 数据集和 OVERNIGHT 数据集上，我们完整的模型分别取得了 85.5 和 79.0（八个领域的平均准确率）的准确率，都是当前最好的结果，甚至比有利用额外训练语料的 Jia and Liang [68]* 结果要好。

2) 相比之前的序列-序列（Seq2Seq）的模型中所使用的序列化逻辑表达式，我们所使用的动作序列编码更加有效。在三个数据集上，我们最基本的系统 Seq2Act 的结果比 Seq2Seq 模型都要好。在 GEO 数据集上，Seq2Act 的准确率是 87.5，而 Seq2Seq 模型中最好的结果是 87.2；在 ATIS 数据集，Seq2Act 的准确率是 84.6，与 Seq2Seq 模型中最好的结果一样；在 OVERNIGHT 数据集上，Seq2Act 的准确率是

	Soc.	Blo.	Bas.	Res.	Cal.	Hou.	Pub.	Rec.	Avg.
Previous Work									
Wang et al. [139]	48.2	41.9	46.3	75.9	74.4	54.0	59.0	70.8	58.8
Seq2Seq Models									
Xiao et al. [144]	80.0	55.6	80.5	80.1	75.0	61.9	75.8	—	72.7
Jia and Liang [68]	81.4	58.1	85.2	76.2	78.0	71.4	76.4	79.6	75.8
Jia and Liang [68]* (+data)	79.6	60.2	87.5	79.5	81.0	72.5	78.3	81.0	77.5
Our Models									
Seq2Act	81.4	60.4	87.5	79.8	81.0	73.0	79.5	81.5	78.0
Seq2Act (+C1)	81.8	60.9	88.0	80.1	81.0	73.5	80.1	82.0	78.4
Seq2Act (+C1+C2)	82.1	61.4	88.2	80.7	81.5	74.1	80.7	82.9	79.0

表 5.2: OVERNIGHT 数据集上的测试结果

78.0, 而 Seq2Seq 模型中最好的结果是 77.5。我们认为这是因为我们所采用的动作序列编码更紧凑, 另外能够编码更多的信息, 特别是同时编码句法信息和语义信息(这点与 Zettlemoyer and Collins [154] 所使用的基于 CCG 的词汇有点类似)。

3) 句法约束条件能够基于保证语义图结构有效原则对候选动作进行刷选, 及时过滤违背该约束的动作, 这样一来能够有效提高语义解析的性能。在三个数据集上, 有使用句法约束条件的系统 Seq2Act (+C1) 比最基本的系统 Seq2Act 都有所提高。

4) 语义约束条件能够进一步的提高语义解析的性能。在三个数据集上, 我们完整的模型 Seq2Act (+C1+C2) 的结果都超过了 Seq2Act 和 Seq2Act (+C1) 的结果。我们认为这是因为我们利用选择约束和类别不冲突约束, 能够在解码的时候, 及时过滤掉违背这些约束条件的错误的动作候选, 从而提高了最终结果是准确的可能性。

5.4.3 详细分析

对比两种实体处理机制 在前面章节已经提到, 我们实现了两种实体处理机制, 一种是基于替代 (*Replacing*) 的机制 [43]: 把实体替换成实体的类别加上独有的 ID; 另一种是基于复制 (*Copying*) 的机制 [68]。为了对比两种实体处理机制, 我们用我们的完整模型分别利用两种实体处理机制进行了对比实验, 实验结果如表5.3所示。我们可以看到在三个数据集上, 基于 *Replacing* 的实体处理机制比基于 *Copying* 的实体处理机制的效果都好。我们认为这是因为 *Replacing* 是在预处理中处理的, 能够有效消除由实体带来的稀疏问题; 而 *Copying* 是存在于训练和测试中的, 并且需要额外的 *Copying* 机制, 详情参见 [68]。

	<i>Replacing</i>	<i>Copying</i>
GEO	88.9	88.2
ATIS	85.5	84.0
OVERNIGHT	79.0	77.9

表 5.3: 在三个数据集上对比两种实体处理机制的结果

	Logical Form	Action Sequence
GEO	28.2	18.2
ATIS	28.4	25.8
OVERNIGHT	46.6	33.3

表 5.4: 在三个数据集上线性化逻辑表达式和动作序列的平均长度

对比线性化的逻辑表达式和动作序列 之前的 Seq2Seq 模型直接使用线性化的逻辑表达式作为目标端的序列，我们使用动作序列作为目标端的序列，表5.4中显示了两者的平均长度。我们可以看到，动作序列比线性化的逻辑表达式更加的紧凑：相比线性化的逻辑表达式，动作序列更短，在三个数据集上，长度分别减少了 35.5%、9.2% 和 28.5%。在 OVERNIGHT 数据集上，我们对 8 个领域分别统计，在进行平均。这种更短/更紧凑的编码方式能够稍微减缓一下目标端长距离依赖的问题，并且也有利于学习自然语言句子与目标端的语义表示之间的对齐。

5.4.4 错误分析

我们在错误的样例上进行了错误分析，我们发现有两类错误比较显著，也应该引起我们的注意。我们在表5.5中展示了几个错误样例，每个例子由三部分组成：句子、正确的逻辑表达式和我们模型预测的逻辑表达式，另外我们用红颜色标记了错误的地方。这两类错误如下：

未出现/不正式的句法结构 有些测试样例有没有在语料中出现过的句法结构。表的第一个例子中，句子的这种句法结构并不常见，实体词 “Iowa” 和关系词 “borders” 都出现在疑问词 “how many” 之前，正常的形式应该是：“How many states does Iowa border?”。对于这种问题，我们可以尝试利用句子重写或者复述 [24, 45] 的方法把有不常见句法结构的句子转换成有常见句法结构的句子。

匹配遗漏 (Under-Mapping) 如 Dong and Lapata [43]讨论的，像我们所使用的基于注意力机制的循环神经网络模型并没有考虑对齐的历史，当前的注意力分布并没有把之前的注意力分布考虑进行，这样容易造成源端的有些词语并没有被注意力集中到，从而导致某些词语没有匹配到，出现匹配遗漏的情况，如表中的第

Error Types	Examples
Un-covered Sentence Structure	Sentence: <i>Iowa borders how many states?</i> (Formal Form: <i>How many states does Iowa border?</i>) Gold Parse: <code>answer(A, count(B, (const (C, stateid(iowa)), next_to(C, B), state (B)), A))</code> Predicted Parse: <code>answer (A, count(B, state(B), A))</code>
Under-Mapping	Sentence: <i>Please show me first class flights from indianapolis to memphis one way leaving before 10am</i> Gold Parse: <code>(lambda x (and (flight x) (oneway x) (class_type x first:cl) (< (departure_time x) 1000:ti) (from x indianapolis:ci) (to x memphis:ci)))</code> Predicted Parse: <code>(lambda x (and (flight x) (oneway x) (< (departure_time x) 1000:ti) (from x indianapolis:ci) (to x memphis:ci)))</code>

表 5.5: 用于错误分析的几个样例

二个例子，“*first class*”就被忽略了，本来它是要映射到 `(class_type x first:cl)` 的。这类问题在神经机器翻译中也很常见，类似的，我们可以利用注意力覆盖度的机制 [134, 33] 来处理这种问题。

5.5 本章小结

本章同时从语义落地和结构预测两个方面来研究语义解析。针对现有语义解析方法过于依赖于高质量词典、特定文法和人工定义特征的问题，我们提出了一种端到端语义图生成的语义解析方法。该方法同时利用了语义图的表示优势，和循环神经网络模型的强序列预测能力。具体的，我们将语义解析问题转换为自然语言句子序列到语义图构建的动作序列的翻译问题，我们使用循环神经网络模型来建模动作序列的生成。相比于传统语义解析方法，该方法使用语义图来表示句子的语义，不再需要特定文法；相比于新兴的语义解析方法，该方法使用动作序列来编码语义图的构建，循环神经网络模型来建模动作序列的生成，该过程不再是启发式，是端到端的，同时在解析的过程中能够考虑到动作与动作之间的相互联系，能够方便地加入句法和语义约束条件，进一步提升语义解析的准确度。最后，我们在三个公开数据集上进行了实验，实验结果论证了我们方法的有效性：我们的方法在其中两个数据集上都取得了当前最好的效果，在其中一个数据集上也取得了具有竞争力的结果。

本章的相关研究成果 “Sequence-to-Action: End-to-End Semantic Graph

Generation for Semantic Parsing” 已被 ACL-2018 录用（长文）。

第六章 总结与展望

6.1 工作总结

随着人工智能的热潮，自然语言处理领域受到了人们更多的关注。现在自然语言处理的各个任务，小到分词，大到句法分析，机器翻译都在尝试利用更多更准确的语义信息。产业界对人机交互的需求也在持续增长，迫切需要自然语言理解技术，而自然语言理解的核心之一就是语义解析。

语义解析是一项富有挑战的研究任务。语义解析需要把自然语言句子解析成计算机可识别的语义表示，相当于把自然语言映射成计算机语言，而计算机语言是一门严格的形式化语言，人类的自然语言形式则变化多端，表达也丰富多彩，一个意思可以有多种表达，从词汇级别的变化再到句法结构上的变化，还有口头语和书面语的不同，以及不同社交媒体上的不同风格的语言。这都使得上述映射过程十分困难。

语义解析的发展过程中，研究者提出了两个理论：组合语义学（compositional semantics）和模型理论语义学（model-theoretic semantics），这也成了现阶段语义解析的理论基础，同时也为语义解析指出了发展路线：语义解析需要进行结构预测和语义落地。具体的，组合语义学指出句子的意思是句子中各成分的意思的组合，表明句子最终的语义表示是具有结构的，结构预测指的就是预测句子所对应的语义表示的结构。模型理论语义学指出要用世界中的状态来解释句子，表明需要将句子中的具体的项对应到世界中实际的项（实体和关系），这就是语义落地过程。

本文从语义解析的两个核心部分：结构预测和语义落地出发，对语义解析技术展开研究。针对上述两个核心部分所存在的挑战，以及现有方法存在的缺陷，本文尝试提出可行的解决思路。现将本论文的主要工作和成果总结如下：

- 针对语义解析中词典覆盖度不够的问题，提出了两种基于词典扩充学习的语义解析方法，一种是在现有工作的基础上，在训练的时候引入新的词汇，同时加以学习，为了得到准确度更高的词汇，在词典扩充学习的基础上设计了通用的词语-谓词特征模板，还提出了基于投票机制的核心词典获取方法。另一种利用大量文本语料和词典资源学习词语之间的相似度，再从少量种子词汇出发，利用标签传播算法，把标注节点的谓词映射信息传播给未标注的节点，从而学习出更多的新词汇，相比于之前的方法，这种方法不仅利用知识库与文本之间的实体对共现来学习初始词汇，还充分利用了大量自由文本和收集好的词典资源来学习词语与词语之间的语义关联，并使用基于图的半监督学习框架来进行词典扩充学习。实验结果验证了我们词典扩充学习方法的

有效性，通过词典的扩充学习，词典的覆盖度得以提升，具体体现在基于扩充词典的召回率显著提升。

- 针对语义解析中句子的结构与目标语义表示的结构不一致的问题，提出了一种基于句子重写的语义解析方法。通过对句子不一致问题的分析和归纳，总结出了两种结构不一致的问题：1-N型不匹配和N-1型不匹配，进而对应地提出了两种句子重写方法，分别是基于词典替换的句子重写和基于模板替换的句子重写。这个方法的出发点是一个句子总会存在另外一种表述形式，该形式保持与原句子的意思相同，并且结构与目标语义表示的结构一致。我们将句子重写运用于语义解析中，具体的，对于给定的句子，先通过句子重写，得到若干新的句子，再利用基准语义解析器对每一个新句子进行解析，得到候选逻辑表达式，我们定义联合打分函数来进行解码，该打分函数综合考虑句子重写部分的特征和语义解析部分的特征。实验结果验证了我们句子重写方法的有效性，通过句子重写，使得语义解析器能够解析存在结构不匹配的句子。
- 针对传统语义解析方法过于依赖词典和特定文法、模板、依存树或者启发式规则的问题，提出了一种基于端到端语义图生成的语义解析方法。该方法综合利用了语义图的语义表示能力，以及循环神经网络模型的强预测能力。具体的，我们用语义图来表示句子的语义，把语义解析过程看作是语义图构建的过程，同时使用动作序列来编码语义图的构建，这样一来语义解析就转换为句子到动作序列的映射，我们再利用循环神经网络模型来建模这个过程，且在解码的过程中，我们可以加入句法约束条件和语义约束条件来过滤错误的动作，从而提高语义解析的准确度。相比之前的基于语义图的方法，该方法不依赖文法、模板、依存树和启发式规则，整个模型的学习是端到端的，适用性更强。相比之前的序列-到-序列的方法，该方法使用语义图来表示语义，动作序列的编码方式能够建模更多的句法信息，同时还可以考虑解码时，当前动作与已生成动作之间存在联系，可以利用句法约束和语义约束来提高解码的准确性。实验结果验证了我们方法的有效性。

6.2 工作展望

语义解析在自然语言处理中有着重要的地位，成熟的语义解析技术能对自然语言处理领域中的其他任务产生积极的推进作用，如机器翻译，信息抽取等。在产业界，语义解析技术也有着很高的应用价值，能够应用于智能机器人，智能搜索引擎等。虽然近年来语义解析技术有了稳定的发展，本文在语义解析方面也取得了一定的研究成果，但语义解析离技术广泛运用和推广还有一段距离。本文对未来的语

义解析工作提出以下三点建议和展望：

- 我们可以尝试综合利用基于词典的语义解析方法的优势和基于语义图的语义解析方法的优势。基于词典的语义解析方法以词典为基础，使用自底向上的方式进行解析。只要词典的质量高（包括准确度和覆盖度），就能快速准确的完成对句子的解析。基于语义图的方法可以利用其它成熟的手段（如句法分析，依存分析）先确定句子的结构，即语义表示的框架，再利用先进的技术手段（如神经网络模型）进行实体链接和关系匹配，整个过程类似于以自顶向下的方式进行。我们可以尝试设计基于混合策略的解析算法，其能够同时利用两类方法的优势。
- 目前我们所使用的句子重写方法还过于简单，都是启发式的方法，需要借助人工收集的本文语料和文本资源，将来我们可以尝试使用更加复杂的方法，同时减少对人工语料的依赖，如利用神经网络模型从大量自动抽取的文本中，学习句子重写的方法。
- 针对语义解析中使用逻辑表达式进行标注的训练语料严重不够的问题，一方面，我们可以尝试使用基于句子-答案的语料来训练，这还需要设计远距离监督学习算法。另外一方面，我们还可以尝试利用语义图来标注句子的语义，再从语义图转换为逻辑表达式。

参考文献

- [1] Alfred V. Aho and Jeffrey D. Ullman. *The Theory of Parsing, Translation, and Compiling*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1972.
- [2] Jacob Andreas, Andreas Vlachos, and Stephen Clark. Semantic parsing as machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 47–52, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P13-2009>.
- [3] Yoav Artzi and Luke Zettlemoyer. Bootstrapping semantic parsers from conversations. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 421–432, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D11-1039>.
- [4] Yoav Artzi and Luke Zettlemoyer. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1(1):49–62, 2013.
- [5] Yoav Artzi, Dipanjan Das, and Slav Petrov. Learning compact lexicons for ccg semantic parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1273–1283, Doha, Qatar, October 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D14-1134>.
- [6] Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. Broad-coverage ccg semantic parsing with amr. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL <http://aclweb.org/anthology/D15-1198>.
- [7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL <http://arxiv.org/abs/1409.0473>.
- [8] Collin Baker. Framenet: A knowledge base for natural language processing. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*, pages 1–5. Association for Computational Linguistics, 2014. URL <http://www.aclweb.org/anthology/W14-3001>.
- [9] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project. In *COLING-ACL '98: Proceedings of the Conference*, pages 86–90, Montreal, Canada, 1998.

- [10] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186. Association for Computational Linguistics, 2013. URL <http://www.aclweb.org/anthology/W13-2322>.
- [11] Junwei Bao, Nan Duan, Ming Zhou, and Tiejun Zhao. Knowledge-based question answering as machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 967–976, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P14-1091>.
- [12] Hannah Bast and Elmar Haussmann. More accurate question answering on freebase. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia*, pages 1431–1440, 2015. URL <http://doi.acm.org/10.1145/2806416.2806472>.
- [13] I. Beltagy and Chris Quirk. Improved semantic parsers for if-then statements. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 726–736, Berlin, Germany, August 2016. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P16-1069>.
- [14] Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P14-1133>.
- [15] Jonathan Berant and Percy Liang. Imitation learning of agenda-based semantic parsers. *Transactions of the Association for Computational Linguistics*, 3:545–558, 2015.
- [16] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D13-1160>.
- [17] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, pages 1–7, 2010.

- [18] Daniel Bobrow. Natural language input for a computer problem solving system. 01 1964.
- [19] Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620, Doha, Qatar, October 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D14-1067>.
- [20] Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Comput. Linguist.*, 16(2):79–85, 1990. URL <http://dl.acm.org/citation.cfm?id=92858.92860>.
- [21] Qingqing Cai and Alexander Yates. Semantic parsing freebase: Towards open-domain semantic parsing. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 328–338, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S13-1045>.
- [22] Qingqing Cai and Alexander Yates. Large-scale semantic parsing via schema matching and lexicon extension. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 423–433, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P13-1042>.
- [23] Sharon A. Caraballo and Eugene Charniak. New figures of merit for best-first probabilistic chart parsing. *Comput. Linguist.*, 24(2):275–298, 1998. ISSN 0891-2017. URL <http://dl.acm.org/citation.cfm?id=972732.972736>.
- [24] Bo Chen, Le Sun, Xianpei Han, and Bo An. Sentence rewriting for semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 766–777, Berlin, Germany, August 2016. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P16-1073>.
- [25] David L. Chen and Raymond J. Mooney. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI’11, pages 859–865. AAAI Press, 2011. URL <http://dl.acm.org/citation.cfm?id=2900423.2900560>.
- [26] Jianpeng Cheng, Siva Reddy, Vijay Saraswat, and Mirella Lapata. Learning structured natural language representations for semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1)*:

- Long Papers*), pages 44–55, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL <http://aclweb.org/anthology/P17-1005>.
- [27] David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P05-1033>.
- [28] Eunsol Choi, Tom Kwiatkowski, and Luke Zettlemoyer. Scalable semantic parsing with partial ontologies. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1311–1320, Beijing, China, July 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P15-1127>.
- [29] Stephen Clark and James R. Curran. Wide-coverage efficient statistical parsing with ccg and log-linear models. *Comput. Linguist.*, 33(4):493–552, 2007. URL <http://dx.doi.org/10.1162/coli.2007.33.4.493>.
- [30] Stephen Clark, Julia Hockenmaier, and Mark Steedman. Building deep dependency structures using a wide-coverage ccg parser. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 327–334, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P02-1042>.
- [31] James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. Driving semantic parsing from the world’s response. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 18–27, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W10-2903>.
- [32] John Cocke. *Programming Languages and Their Compilers: Preliminary Notes*. Courant Institute of Mathematical Sciences, New York University, 1969.
- [33] Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885, San Diego, California, June 2016. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N16-1102>.
- [34] Michael Collins. Head-driven statistical models for natural language parsing. *Comput. Linguist.*, 29(4):589–637, 2003. URL <http://dx.doi.org/10.1162/08912010322753356>.
- [35] Michael Collins, Philipp Koehn, and Ivona Kucerova. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of*

- the Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P05-1066>.
- [36] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, 2011. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=1953048.2078186>.
- [37] Rodolfo Corona, Jesse Thomason, and Raymond Mooney. Improving black-box speech recognition using semantic parsing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 122–127, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing. URL <http://www.aclweb.org/anthology/I17-2021>.
- [38] Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. Expanding the scope of the atis task: The atis-3 corpus. In *Proceedings of the Workshop on Human Language Technology*, pages 43–48, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. URL <https://doi.org/10.3115/1075812.1075823>.
- [39] Dipanjan Das and Noah A. Smith. Semi-supervised frame-semantic parsing for unknown predicates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1435–1444, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P11-1144>.
- [40] Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. Probabilistic frame-semantic parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 948–956, Los Angeles, California, June 2010. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N10-1138>.
- [41] Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. Frame-semantic parsing. *Comput. Linguist.*, (1):9–56, 2014. URL http://dx.doi.org/10.1162/COLI_a_00163.
- [42] Donald Davidson. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*. University of Pittsburgh Press, 1967.
- [43] Li Dong and Mirella Lapata. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany, August 2016. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P16-1004>.

- [44] Li Dong, Furu Wei, Ming Zhou, and Ke Xu. Question answering over freebase with multi-column convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 260–269, Beijing, China, July 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P15-1026>.
- [45] Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/D17-1091>.
- [46] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011. URL <http://dl.acm.org/citation.cfm?id=1953048.2021068>.
- [47] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P13-1158>.
- [48] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1156–1165, New York, NY, USA, 2014. ACM. URL <http://doi.acm.org/10.1145/2623330.2623677>.
- [49] Manaal Faruqui, Ryan McDonald, and Radu Soricut. Morpho-syntactic lexicon generation using graph-based semi-supervised learning. *Transactions of the Association for Computational Linguistics*, 4:1–16, 2016.
- [50] Charles J Fillmore. The case for case. In *Proceedings of the Texas Symposium, on Language Universals*, April 1967.
- [51] Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P14-1134>.
- [52] Matthew Gardner and Jayant Krishnamurthy. Open-vocabulary semantic parsing with both distributional statistics and formal knowledge. In *AAAI*, 2017.
- [53] Ruifang Ge and Raymond Mooney. A statistical semantic parser that integrates

- syntax and semantics. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 9–16, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W/W05/W05-0602>.
- [54] Ruifang Ge and Raymond Mooney. Learning a compositional semantic parser using an existing syntactic parser. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 611–619, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P/P09/P09-1069>.
- [55] Ruifang Ge and Raymond J. Mooney. Discriminative reranking for semantic parsing. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 263–270, Sydney, Australia, July 2006. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P/P06/P06-2034>.
- [56] Dan Goldwasser, Roi Reichart, James Clarke, and Dan Roth. Confidence driven unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1486–1495, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P11-1149>.
- [57] William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Austin, Texas, November 2016. Association for Computational Linguistics. URL <https://aclweb.org/anthology/D16-1057>.
- [58] He He, Alvin Grissom II, John Morgan, Jordan Boyd-Graber, and Hal Daumé III. Syntax-based rewriting for simultaneous machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 55–64, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL <http://aclweb.org/anthology/D15-1006>.
- [59] Shizhu He, Kang Liu, Yuanzhe Zhang, Liheng Xu, and Jun Zhao. Question answering over linked data using first-order logic. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1092–1103, Doha, Qatar, October 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D14-1116>.
- [60] Yulan He and Steve Young. Semantic processing using the hidden vector state model. *Computer Speech Language*, 19(1):85 – 106, 2005. URL <http://www.sciencedirect.com/science/article/pii/S0885230804000117>.
- [61] Jonathan Herzig and Jonathan Berant. Neural semantic parsing over multiple

- knowledge-bases. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 623–628, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL <http://aclweb.org/anthology/P17-2098>.
- [62] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.
- [63] Julia Hockenmaier and Mark Steedman. Generative models for statistical parsing with combinatory categorial grammar. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 335–342, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073139. URL <http://www.aclweb.org/anthology/P02-1043>.
- [64] Mark Hopkins, Cristian Petrescu-Prahova, Roie Levin, Ronan Le Bras, Alvaro Herasti, and Vidur Joshi. Beyond sentential semantic parsing: Tackling the math sat with a cascade of tree transducers. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 795–804, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/D17-1083>.
- [65] Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P12-1092>.
- [66] Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. Learning a neural semantic parser from user feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 963–973, Vancouver, Canada, July 2017. Association for Computational Linguistics.
- [67] Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL <http://aclweb.org/anthology/P17-1167>.
- [68] Robin Jia and Percy Liang. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany, August 2016. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P16-1002>.
- [69] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In

- Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D13-1176>.
- [70] Tadao Kasami. An efficient recognition and syntaxanalysis algorithm for context-free languages. page 40, 07 1965.
- [71] Rohit Kate and Raymond Mooney. Semi-supervised learning for semantic parsing using support vector machines. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 81–84, Rochester, New York, April 2007. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N/N07/N07-2021>.
- [72] Rohit J. Kate and Raymond J. Mooney. Learning language semantics from ambiguous supervision. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 1*, AAAI'07, pages 895–900. AAAI Press, 2007. URL <http://dl.acm.org/citation.cfm?id=1619645.1619789>.
- [73] Rohit J. Kate, Yuk Wah Wong, and Raymond J. Mooney. Learning to transform natural to formal languages. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*, pages 1062–1068. AAAI Press, 2005. URL <http://dl.acm.org/citation.cfm?id=1619499.1619504>.
- [74] M Kay. Readings in natural language processing. chapter Algorithm Schemata and Data Structures in Syntactic Processing, pages 35–70. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1986. URL <http://dl.acm.org/citation.cfm?id=21922.24327>.
- [75] Paul Kingsbury and Martha Palmer. From treebank to propbank. In *LREC*, 2002.
- [76] Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. Neural amr: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL <http://aclweb.org/anthology/P17-1014>.
- [77] Jayant Krishnamurthy. Probabilistic models for learning a semantic parser lexicon. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 606–616, San Diego, California, June 2016. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N16-1074>.
- [78] Jayant Krishnamurthy and Thomas Kollar. Jointly learning to parse and perceive: Connecting natural language to the physical world. *TACL*, 1:193–206, 2013.
- [79] Jayant Krishnamurthy and Tom Mitchell. Weakly supervised training of semantic parsers. In *Proceedings of the 2012 Joint Conference on Empirical Methods in*

- Natural Language Processing and Computational Natural Language Learning*, pages 754–765, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D12-1069>.
- [80] Jayant Krishnamurthy and Tom Mitchell. Vector space semantic parsing: A framework for compositional vector space models. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 1–10, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W13-3201>.
- [81] Jayant Krishnamurthy and Tom Mitchell. Learning a compositional semantics for freebase with an open predicate vocabulary. *Transactions of the Association for Computational Linguistics*, 3:257–270, 2015. URL <https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/548>.
- [82] Jayant Krishnamurthy and Tom M. Mitchell. Joint syntactic and semantic parsing with combinatory categorial grammar. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1188–1198, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P14-1112>.
- [83] Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. Neural semantic parsing with type constraints for semi-structured tables. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1526, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/D17-1160>.
- [84] Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. Learning to automatically solve algebra word problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 271–281, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P14-1026>.
- [85] Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1223–1233, Cambridge, MA, October 2010. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D10-1119>.
- [86] Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Lexical generalization in ccg grammar induction for semantic parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1512–1523, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D11-1140>.
- [87] Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. Scaling semantic

- parsers with on-the-fly ontology matching. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1545–1556, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D13-1161>.
- [88] Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. Context-dependent semantic parsing for time expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P14-1135>.
- [89] Junhui Li, Muhua Zhu, Wei Lu, and Guodong Zhou. Improving semantic parsing with enriched synchronous context-free grammar. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1455–1465, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL <http://aclweb.org/anthology/D15-1170>.
- [90] Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23–33, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL <http://aclweb.org/anthology/P17-1003>.
- [91] P. Liang. Lambda dependency-based compositional semantics. *arXiv preprint arXiv:1309.4408*, 2013.
- [92] Percy Liang. Learning executable semantic parsers for natural language understanding. *Commun. ACM*, 59(9):68–76, 2016. URL <http://doi.acm.org/10.1145/2866568>.
- [93] Percy Liang, Michael Jordan, and Dan Klein. Learning dependency-based compositional semantics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 590–599, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P11-1060>.
- [94] Percy Liang, Michael I. Jordan, and Dan Klein. Learning dependency-based compositional semantics. *Comput. Linguist.*, 39(2):389–446, 2013. URL http://dx.doi.org/10.1162/COLI_a_00127.
- [95] Xi Victoria Lin, Chenglong Wang, Luke S. Zettlemoyer, and Michael D. Ernst. Nl2bash: A corpus and semantic parser for natural language interface to the linux operating system. *CoRR*, abs/1802.08979, 2018.
- [96] Wei Lu. Semantic parsing with relaxed hybrid trees. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages

- 1308–1318, Doha, Qatar, October 2014. Association for Computational Linguistics.
URL <http://www.aclweb.org/anthology/D14-1137>.
- [97] Wei Lu. Constrained semantic forests for improved discriminative semantic parsing. In *ACL (2)*, pages 737–742, 2015.
- [98] Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke S. Zettlemoyer. A generative model for parsing natural language to meaning representations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 783–792, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D08-1082>.
- [99] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL <http://aclweb.org/anthology/D15-1166>.
- [100] Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. Learning to parse natural language to a robot execution system, 2012.
- [101] Ramtin Mehdizadeh Seraj, Maryam Siahbani, and Anoop Sarkar. Improving statistical machine translation with a multilingual paraphrase database. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1379–1390, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL <http://aclweb.org/anthology/D15-1163>.
- [102] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. URL <http://arxiv.org/abs/1301.3781>.
- [103] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’13, pages 3111–3119, USA, 2013. Curran Associates Inc. URL <http://dl.acm.org/citation.cfm?id=2999792.2999959>.
- [104] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N13-1090>.
- [105] Marvin Minsky. A framework for representing knowledge. Technical report, Cambridge, MA, USA, 1974.
- [106] Dipendra Kumar Misra and Yoav Artzi. Neural shift-reduce ccg semantic parsing. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language*

- Processing*, pages 1775–1786, Austin, Texas, November 2016. Association for Computational Linguistics. URL <https://aclweb.org/anthology/D16-1183>.
- [107] R. Montague. Universal grammar. *Theoria*, 36(3):373–398, 1970.
- [108] Richard Montague. English as a formal language. In Bruno Visentini, editor, *Linguaggi nella societa e nella tecnica*, pages 188–221. Edizioni di Communita, 1970.
- [109] Richard Montague. The proper treatment of quantification in ordinary english. In Patrick Suppes, Julius Moravcsik, and Jaakko Hintikka, editors, *Approaches to Natural Language*, pages 221–242. Dordrecht, 1973.
- [110] Raymond J. Mooney. Learning semantic parsers: An important but under-studied problem. In *In AAAI 2004 Spring Symposium on Language Learning: An Interdisciplinary Perspective*, pages 39–44, 2004.
- [111] Shashi Narayan, Siva Reddy, and Shay B. Cohen. Paraphrase generation from latent-variable pcfgs for semantic parsing. *CoRR*, abs/1601.06068, 2016. URL <http://arxiv.org/abs/1601.06068>.
- [112] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1):71–106, 2005. URL <http://dx.doi.org/10.1162/0891201053630264>.
- [113] Ankur P. Parikh, Hoifung Poon, and Kristina Toutanova. Grounded semantic parsing for complex knowledge extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 756–766, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N15-1077>.
- [114] Terence Parsons. *Events in the semantics of English: a study in subatomic semantics*. MIT Press, Cambridge, MA and London, 1990.
- [115] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China, July 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P15-1142>.
- [116] Adam Pauls and Dan Klein. K-best a* parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 958–966, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P/P09/P09-1108>.
- [117] Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb 2.0: Better paraphrase ranking, fine-grained entail-

- ment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China, July 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P15-2070>.
- [118] Hoifung Poon. Grounded unsupervised semantic parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 933–943, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P13-1092>.
- [119] Hoifung Poon and Pedro Domingos. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Singapore, August 2009. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D/D09/D09-1001>.
- [120] Michael Pust, Ulf Hermjakob, Kevin Knight, Daniel Marcu, and Jonathan May. Parsing english into abstract meaning representation using syntax-based machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1143–1154, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL <http://aclweb.org/anthology/D15-1136>.
- [121] Chris Quirk, Raymond Mooney, and Michel Galley. Language to code: Learning semantic parsers for if-this-then-that recipes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 878–888, Beijing, China, July 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P15-1085>.
- [122] Majid Razmara, Maryam Siahbani, Reza Haffari, and Anoop Sarkar. Graph propagation for paraphrasing out-of-vocabulary words in statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P13-1109>.
- [123] Siva Reddy, Mirella Lapata, and Mark Steedman. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*, 2:377–392, 2014. URL <http://aclweb.org/anthology/Q14-1030>.
- [124] Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. Transforming Dependency Structures to Logical Forms for Semantic Parsing. *Transactions of the Association for Computa-*

- tional Linguistics*, 4:127–140, 2016. ISSN 2307-387X. URL <http://sivareddy.in/papers/reddy2016transforming.pdf>.
- [125] Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. Universal semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 89–101, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/D17-1009>.
- [126] Avneesh Saluja, Hany Hassan, Kristina Toutanova, and Chris Quirk. Graph-based semi-supervised learning of translation models from monolingual data. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 676–686, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P14-1064>.
- [127] M. Steedman. *Surface Structure and Interpretation*. MIT Press, 1996.
- [128] Mark Steedman. *The Syntactic Process*. MIT Press, Cambridge, MA, USA, 2000.
- [129] Huan Sun, Hao Ma, Xiaodong He, Wen-tau Yih, Yu Su, and Xifeng Yan. Table cell search for question answering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW ’16, pages 771–782, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee. URL <https://doi.org/10.1145/2872427.2883080>.
- [130] Raymond Hendy Susanto and Wei Lu. Neural architectures for multilingual semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 38–44, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL <http://aclweb.org/anthology/P17-2007>.
- [131] Partha Pratim Talukdar and Koby Crammer. New regularized algorithms for transductive learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 442–457. Springer, 2009.
- [132] Lappoon R. Tang and Raymond J. Mooney. Using multiple clause constructors in inductive logic programming for semantic parsing. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL ’01, pages 466–477, London, UK, 2001. Springer-Verlag. URL <http://dl.acm.org/citation.cfm?id=645328.650015>.
- [133] Cynthia A. Thompson and Raymond J. Mooney. Corpus-based lexical acquisition for semantic parsing. 1996.
- [134] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

- 76–85, Berlin, Germany, August 2016. Association for Computational Linguistics.
URL <http://www.aclweb.org/anthology/P16-1008>.
- [135] Adrienne Wang, Tom Kwiatkowski, and Luke Zettlemoyer. Morpho-syntactic lexical generalization for ccg semantic parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1284–1295, Doha, Qatar, October 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D14-1135>.
- [136] Chuan Wang, Nianwen Xue, and Sameer Pradhan. A transition-based algorithm for amr parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N15-1040>.
- [137] S. I. Wang, P. Liang, and C. Manning. Learning language games through interaction. In *Association for Computational Linguistics (ACL)*, 2016.
- [138] Ye-Yi Wang and Alex Waibel. Modeling with structures in statistical machine translation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 1357–1363, Montreal, Quebec, Canada, August 1998. Association for Computational Linguistics. doi: 10.3115/980691.980790. URL <http://www.aclweb.org/anthology/P98-2221>.
- [139] Yushi Wang, Jonathan Berant, and Percy Liang. Building a semantic parser overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1332–1342, Beijing, China, July 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P15-1129>.
- [140] Yuk Wah Wong and Raymond Mooney. Learning for semantic parsing with statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 439–446, New York City, USA, June 2006. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N/N06/N06-1056>.
- [141] Yuk Wah Wong and Raymond Mooney. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 960–967, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P07-1121>.
- [142] William Woods, Ronald Kaplan, and Bonnie Webber. The lunar science natural language information system: Final report. 01 1972.

- [143] Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist.*, 23(3):377–403, 1997. URL <http://dl.acm.org/citation.cfm?id=972705.972707>.
- [144] Chunyang Xiao, Marc Dymetman, and Claire Gardent. Sequence-based structured prediction for semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1341–1350, Berlin, Germany, August 2016. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P16-1127>.
- [145] Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. Question Answering on Freebase via Relation Extraction and Textual Evidence. In *Proceedings of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany, August 2016. Association for Computational Linguistics. URL <http://sivareddy.in/papers/kun2016question.pdf>.
- [146] Min-Chul Yang, Nan Duan, Ming Zhou, and Hae-Chang Rim. Joint relational embeddings for knowledge-based question answering. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 645–650, Doha, Qatar, October 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D14-1071>.
- [147] Xuchen Yao. Lean question answering over freebase from scratch. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 66–70, Denver, Colorado, June 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N15-3014>.
- [148] Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 956–966, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P14-1090>.
- [149] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China, July 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P15-1128>.
- [150] Daniel H. Younger. Recognition and parsing of context-free languages in time n^3 . *Information and Control*, 10(2):189 – 208, 1967. URL <http://www.sciencedirect.com/science/article/pii/S001999586780007X>.
- [151] John M. Zelle and Raymond J. Mooney. Learning to parse database queries us-

- ing inductive logic programming. In *AAAI/IAAI*, pages 1050–1055, Portland, OR, August 1996. AAAI Press/MIT Press. URL <http://www.cs.utexas.edu/users/ai-lab/?zelle:aaai96>.
- [152] Luke Zettlemoyer and Michael Collins. Online learning of relaxed CCG grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 678–687, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D/D07/D07-1071>.
- [153] Luke Zettlemoyer and Michael Collins. Learning context-dependent mappings from sentences to logical form. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 976–984, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL [http://www.aclweb.org/anthology/P/P09-1110](http://www.aclweb.org/anthology/P/P09/P09-1110).
- [154] Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005*, pages 658–666, 2005.
- [155] Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. A tree sequence alignment-based tree-to-tree translation model. In *Proceedings of ACL-08: HLT*, pages 559–567, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P/P08/P08-1064>.
- [156] Yuchen Zhang, Panupong Pasupat, and Percy Liang. Macro grammars and holistic triggering for efficient semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1214–1223, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/D17-1125>.
- [157] Kai Zhao, Hany Hassan, and Michael Auli. Learning translation models from monolingual continuous representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1527–1536, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N15-1176>.
- [158] Lei Zou, Ruizhe Huang, Haixun Wang, Jeffrey Xu Yu, Wenqiang He, and Dongyan Zhao. Natural language question answering over rdf: A graph data driven approach. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management*

- of Data*, SIGMOD '14, pages 313–324, New York, NY, USA, 2014. ACM. URL <http://doi.acm.org/10.1145/2588555.2610525>.
- [159] 冯志伟. 自然语言的计算机处理. 上海外语教育出版社, 1996.
- [160] 宗成庆. 统计自然语言处理. 清华大学出版社, 2013.
- [161] 张强. 基于机器翻译的中文语义解析. 东南大学硕士学位论文, 2015.
- [162] 郑晓东, 胡汉辉, 赵林度, and 吕永涛. 一种基于词对齐的中文深层语义解析模型. 计算机工程与应用, pages 8–13, 2017.

简历

基本情况

陈波，男，湖南长沙宁乡人，1988 年出生，中国科学院软件研究所博士研究生。

教育状况

2012 年 09 月至今，就读于中国科学院软件研究所，中文信息处理实验室，硕博连读生，2014 年转博；

2008 年 09 月至 2012 年 06 月，就读于中国科学技术大学，计算机科学与技术专业，获得学士学位。

联系方式

通讯地址：北京市海淀区中关村南四街 4 号 5 号楼

邮编：100190

E-mail: chenbo@nfs.iscas.ac.cn

研究兴趣

自然语言处理，语义解析

攻读博士学位期间参与项目

[1] 863 课题：面向基础教育的知识关联与推理类问题求解关键技术与系统（项目编号：2015AA015405）

[2] 国家自然科学基金重点课题：汉语认知加工机制与计算模型（项目编号：61433015）

[3] 中国科协项目：中国科协国际科技组织及任职数据库查询系统

攻读博士学位期间发表的学术论文

◎ 已发表或已录用的论文：

- [1] **Bo Chen**, Le Sun, Xianpei Han, Bo An. Sentence Rewriting for Semantic Parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL-2016)*. Association for Computational Linguistics, Berlin, German, 2016, pages 766-777. (长文, 已发表, CCF-A 类)
- [2] **Bo Chen**, Bo An, Le Sun, Xianpei Han. Sequence-to-Action: End-to-End Semantic Graph Generation for Semantic Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL-2018)* (长文, 已录用, CCF-A 类)
- [3] 陈波, 孙乐, 韩先培. 基于桥连接的词典学习方法的语义解析. 中文信息学报. 2018. (已录用)
- [4] Bo An, **Bo Chen**, Xianpei Han, Le Sun. Accurate Text-Enhanced Knowledge Graph Representation Learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-2018)*. (长文, 已录用, CCF-C 类)
- [5] Kees van Deemter, Le Sun, Rint Sybesma, Xiao Li, **Bo Chen**, Muyun Yang. Investigating the content and form of referring expressions in Mandarin: introducing the Mtuna corpus. In *Proceedings of the 10th International Conference on Natural Language Generation (INLG-2017)*. Association for Computational Linguistics, Santiago de Compostela, Spain, 2017, pages 213-217. (短文, 已发表)

◎ 已投稿的论文：

- [1] **Bo Chen**, Bo An, Le Sun, Xianpei Han. Semi-Supervised Lexicon Learning for Wide-Coverage Semantic Parsing. (长文, 已投 Coling-2018)

◎ 已申请的国家发明专利：

- [1] 孙乐, 韩先培, 陈波. 基于句子重写的语义分析方法及装置. 受理号:201611046181.7
- [2] 陈波, 孙乐, 韩先培. 基于半监督学习的语义分析方法及系统. 受理号:201810054918.2

致 谢

转眼之间，我在中国科学院软件研究所读书已有 6 年，还记得 2012 年 8 月的一天，我和我的两位同学一起坐火车来北京求学，那时的我对未来是既迷茫又期待，只知道自己可能会走上一条科研的路，但这条路会是一个什么样子，我是否能够一直走下去，这些都是未知数。2014 年转博，那时的我已经坚定了科研的信念，也憧憬着接下来几年的科研生活，畅想着写完博士大论文的那一天，会在致谢里面写上什么。此刻，当年畅想的这天已经来临，大论文终于完成。细细回顾自己的研究生生活，想到的只有感激和感恩，在此谨向所有给予我关心和帮助的老师、同学、朋友和家人表示最衷心的感谢！

首先，我要诚挚的感谢我的导师孙乐研究员。进实验室的第一个月里，孙老师就给我指明了一个极具前瞻性又富有挑战的研究方向——语义解析，也给我推荐了第一篇论文——斯坦福的 Percy Liang 发表在 ACL-2011 上的一篇论文，并要求我一周后在组会上作论文报告。那时的我对科研一无所知，只知道借助翻译来阅读论文，当年 Percy Liang 的博士论文只选择了这一个工作，足可见这篇论文的贡献之大，但也反映了这篇论文的阅读难度，特别是对于当年就是科研小白的我。后来的组会报告效果可想而知，大家似乎并不明白我在讲什么，其实我应该也没有理解论文讲了什么。事后孙老师教我如何读论文，如何跟进自己领域最前沿的研究状况。这些都为我以后的学习和科研打下了夯实的基础。学术上，孙老师一直教导我坚守自己的研究内容，坚定发现问题-解决问题的研究思路，在我每次踌躇不前的时候，孙老师总是找我交流，给我提供重要的指导意见。生活上，孙老师也是给我提供所有可能的帮助，凡事都站在我的角度思考，给我分享人生的经验，在我走弯路的时候，及时把我拉回正轨。正是孙老师渊博的知识、严谨的科学态度和正派的为人处世，让我的学习和生活都得到了巨大的帮助。

我还要特别感谢实验室的老师韩先培副研究员，我的每一篇论文都离不开韩老师的仔细打磨。韩老师超高的学术素养给我的科研带来很大的帮助和影响，韩老师教我如何提炼问题，如何提炼方法，如何撰写科技论文。每次帮我改论文的时候，都是严格而细心，不放过任何一个不妥的地方，还教我如何保持论文的逻辑性，现在回想起来只有满满的感动和感激。另外韩老师踏实做事，为人友善的风格，也让我受益良多。

感谢实验室的老师吴健研究员、马龙龙副研究员、刘汇丹副研究员、何苯副研究员，感谢他们给我无私的帮助和关心。感谢我的师兄张振中博士、石贝博士，感谢他们在我刚进实验室的时候，给予我科研上的指导和生活上的帮助。感谢我的师弟宋希良、颜令勇、朱其琛、林鸿宇、刘方超、陆垚杰、刘泽宇和金琪，还有在读博士生安波和付成，平时和他们一起参加论文讨论会，和他们的交流给了我许多启

发。感谢实验室的刘净老师和陈雷诺先生，感谢他们在生活上给予我的帮助。

我要真诚的感谢我的家人，感谢我的爸爸妈妈提供给我的学习条件，让我能一直念到博士，整整 23 年的读书生涯，离不开他们在背后默默的付出，正是他们的支持和鼓励，才能让我顺利完成学业。也要感谢我的弟弟和弟媳，感谢他们对家庭的支持和付出，让我可以集中精力在学业上。在此，特别感谢我的妻子邓新星女士，感谢她的理解和支持，在我一无所有的时候毫无怨言的陪伴着我，无条件支持我的学业，给予我鼓励，让我能够安心科研。我要感谢我所有的亲人，感谢他们在我成长的过程中给予我关怀，让我有一个幸福的大家庭。也感谢我的岳父岳母，感谢他们在生活上给我无私的关怀和帮助。

最后，对所有参加我博士学位论文评审、答辩的老师表示由衷的感谢！