中图分类号：TP391
论 文 编 号：10006BY1706123

北京航空航天大學

博 士 学 位 论 文

弱标注下的自然语言语义解析技术研究

作 者 姓 名　刘乾
学 科 专 业　计算机应用技术
指 导 教 师　赵沁平　教授
　　周彬　副教授
培 养 学 院　计算机学院

Semantic Parsing of Natural Language from Weakly Labeled Data

A Dissertation Submitted for the Degree of Doctor of Philosophy

Candidate: **LIU Qian**

Supervisor: **Professor ZHAO Qinping**

Associate Professor ZHOU Bin

School of Computer Science and Engineering

Beihang University, Beijing, China

中图分类号：TP391
论 文 编 号：10006BY1706123

博 士 学 位 论 文

弱标注下的自然语言语义解析技术研究

作者姓名	刘乾	申请学位级别	工学博士
指导教师姓名	赵沁平 周彬	职 称	教授 副教授
学科专业	计算机应用技术	研究方向	自然语言处理
学习时间自	2017 年 09 月 01 日	起 至	2022 年 06 月 04 日止
论文提交日期	2022 年 05 月 10 日	论文答辩日期	2022 年 06 月 04 日
学位授予单位	北京航空航天大学	学位授予日期	年 月 日

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：_____ 日期： 年 月 日

学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名：_____ 日期： 年 月 日

指导教师签名：_____ 日期： 年 月 日

摘　要

自然语言语义解析，旨在构建能将用户输入的自然语言解析成形式化的计算机程序并完成下游复杂任务的机器，是自然语言处理领域内最具挑战的方向之一。一方面，语义解析是理解人类语言道路上的一大里程碑，因此具有重大的研究价值。另一方面，语义解析允许没有任何编程背景的用户用自然语言完成原本需要复杂程序才能完成的任务，因此具有很高的商业价值。然而，语义解析中的程序数据标注需要耗费大量的人力和财力，阻碍了该领域的规模化发展，因此开展弱标注下的自然语言语义解析方法与关键技术研究迫在眉睫。弱标注意味着数据数量和质量两个维度的弱化，这为语义解析研究提出了极大挑战。当方法能使用的数据数量较弱时，模型面临程序标注规模小和领域标注种类少的难题。当方法能使用的数据质量较弱时，模型面临答案标注监督弱和对话标注构造难的难题。

本论文围绕自然语言语义解析方向，调研分析了国内外相关研究现状与发展趋势，针对弱标注下自然语言语义解析所面临的程序标注规模小、领域标注种类少、答案标注监督弱和对话标注构造难等难题，从模型面向程序组合泛化能力提升，模型面向知识库领域泛化能力提升，弱监督下答案驱动语义解析模型性能提升，半监督下对话重写驱动的对话式语义解析模型等方面开展若干关键技术研究。本文的主要创新点如下：

1、针对程序标注规模小的难点，为增强语义解析模型面向程序组合泛化的能力，本文提出一个记忆单元增强的神经网络架构，通过多步迭代理解自然语言。该架构由寻找局部自然语言片段的组合模块、解析自然语言片段到程序片段的解算模块和存储变量值的记忆单元组成，由它们合作得到自然语言对应的程序。为解决模型优化并缓解奖励稀疏的问题，本文分别提出分层强化学习的算法和课程学习的训练策略。在知名组合泛化评测基准上进行的实验表明，该架构在前人提出的所有组合泛化挑战上表现良好，在所有任务上的准确率均达到 100%。该架构也是首个在全部组合泛化挑战上表现良好的神经网络方法。

2、针对领域标注种类少的难点，为增强语义解析模型面向知识库领域泛化的能力，本文提出一个通过擦除自然语言句单词训练实体链接模型的方法和一个将实体链接结合语义解析模型的框架。无需任何额外标注数据，本方法可以利用预训练语言模型和语义解析数据生成伪标签，并利用这些伪标签学习一个实体链接模型。将该实体链接模型所产生的结果作为先验信息，语义解析模型的解码器可以更好地生成程序。在四个实体链接数据集上的实验结果表明，以伪标签作为监督所训练出的实体链接模型效果远超基

线模型，可取得 7.2% 的准确率提升。在两个领域泛化数据集上的实验结果表明，本方法可以灵活地应用到现有的语义解析模型中，并显著提高它们面向知识库的领域泛化能力，最高可提升 9.8% 的泛化性能。

3、针对答案标注监督弱的难点，为增强语义解析模型在弱监督下的性能，本文提出一个使用生成式语言模型可微分地生成自然语言答案的方法和一个执行引导的预训练方法。与一般预训练需要从互联网爬取预训练语料不同的是，本方法的预训练语料可以自动合成，从而形成高质量的、大规模的预训练语料库。在三个弱监督语义解析数据集上的实验表明，当任务数据较多时，生成式模型性能非常优异，达到了与基线模型持平或更好的性能。当任务数据较少时，执行引导的预训练方法可以显著提升生成式模型的性能，最高可将答案准确率提升 19.5%。最终，本方法在所有实验数据集上均取得最先进的结果，与强监督训练的最好基线模型性能持平。

4、针对对话标注构造难的难点，为增强对话式语义解析模型在半监督下的性能，本文提出将对话式语义解析解耦为对话重写和单轮语义解析两个子任务，从而利用已有的单轮语义解析数据资源。接着，本文构建了首个面向语义解析的对话重写数据集 FollowUp，该数据集由跨 120 张表格的 1,000 个对话组成。为更好地完成对话重写任务，本文提出一个基于拆分重组的对话重写方法，通过直接编辑对话更好地利用对话本身的信息。在 FollowUp 数据集上的实验表明，对话重写任务用于驱动半监督下对话式自然语言语义解析是可行的，且基于拆分重组的方法可以比前人基线模型更好地增强语义解析模型的性能。最终，本方法在半监督下训练的对话式语义解析模型可以达到全监督训练模型 65% 的效果。

关键词：语义解析, 弱监督语义解析, 对话式语义解析, 组合泛化, 领域泛化

Abstract

Semantic parsing over natural language is one of the most challenging directions in the field of natural language processing. It aims to build intelligent machines that can parse natural language questions from users into formal programs, and perform complex downstream tasks. On the one hand, semantic parsing is a major milestone on the road to understanding human language, and therefore has significant research value. On the other hand, semantic parsing allows users without any programming background to accomplish complex tasks with natural language, and thus has significant commercial value. However, program annotation in semantic parsing requires a lot of human and financial resources, which hinders the development of semantic parsing at scale, so it is necessary to carry out research on methods for semantic parsing over natural language under weak annotation. Weak annotation implies a weakening of both data quantity and data quality, which presents a great challenge for the research of semantic parsing. When the data quantity is relatively weak, the model faces the challenges of small program scale and small domain variety. When the data quality is relatively weak, the model faces the challenges of weal answer annotation and difficult conversation annotation construction.

Focusing on the fundamental topics of semantic parsing, we study and review the related works and research trends. And aiming at the challenging problems of small-scale program annotation, small-variety domain annotation, weak supervision of answer annotation and the difficulties of conversation annotation, we propose improving the program-oriented compositional generalization capabilities, improving the knowledge-base domain generalization capabilities, improving the performance of answer-driven semantic parsing model under weak supervision, and rapidly building conversational semantic parsing systems under semi-supervision. The main contributions of this dissertation are summarized as follows.

1. For the difficulty of small-scale program annotation, to enhance the compositional generalization ability of semantic parsing, we propose a memory-augmented neural model, which understand natural language sentences iteratively. The model consists of a Composer module for finding local spans of natural language, a Solver module for parsing natural language spans into program spans and a Memory module for storing variable values. These three modules cooperate to obtain the programs corresponding to natural language sentences. To address model optimization and alleviate the problem of reward sparsity, we propose to employ the hierarchical

reinforcement learning algorithm and the curriculum learning training strategy. Experiments on a typical compositional generalization benchmark show that our model effectively solves all the compositional generalization challenges proposed by previous work, with an accuracy of 100% on all test sets. The model is also the first neural approach that solves all previously proposed compositional generalization challenges.

2. For the difficulty of small-variety domain annotations, in order to enhance the generalization ability of semantic parsing models towards knowledge base domains, we propose a method which trains an entity linking model by erasing words in natural language sentences and a framework which combines entity linking with semantic parsing models. Without any extra labeling effort, our method can automatically generate pseudo-labels with the help of pre-training language models and semantic parsing datasets, and these pseudo-labels can be employed to learn an entity linking model. Regarding entity linking results as the prior, the decoder inside the semantic parser can generate programs more easily. Experimental results on four entity linking datasets show that the entity linking model using pseudo-labels as supervision outperforms baseline model by a large margin, achieving a performance improvement of 7.2%. Experimental results on two semantic parsing datasets for domain generalization show that our method can be flexibly applied to existing semantic parsers and significantly improve their domain generalization capabilities, achieving an absolute improvement of up to 9.8%.

3. For the difficulty of weak answer annotation, in order to enhance the model performance under weak supervision, we propose to first employ generative language models to generate answers for natural language questions, and then perform execution-guided pre-training. Different from the general pre-training methods which require crawling data from the Internet, our pre-training corpus can be synthesized automatically, thus allowing for both large scale and high quality. Experiments on three weakly supervised semantic parsing datasets show that the generative language model can effectively cope with the weakly supervised semantic parsing task when the task data is relatively large, achieving comparable performance to baselines. Additionally, the execution-guided pre-training method can boost the performance of the generative model significantly when the task data is relatively small, achieving an absolute improvement of up to 19.5%. Finally, our method achieves state-of-the-art results on all experimental datasets and can even be comparable with the baseline model under strong supervision.

4. For the difficulty of conversation annotations, in order to enhance the model performance under semi-supervision, we propose to decouple conversational semantic parsing into

two sub-tasks: conversation rewriting and single-round semantic parsing, which enables us to leverage existing single-round semantic parsing datasets. Furthermore, we collect the first conversation rewriting dataset for semantic parsing, FollowUp, which consists of 1,000 conversations across 120 tables. To better accomplish the task of conversation rewriting, we propose a conversation rewriting method based on split-and-recombine, to better utilize the conversation flow by directly editing it. Experiments on the FollowUp dataset show that the conversation rewriting task is feasible for driving semi-supervised conversational semantic parsing, and the split-and-recombine approach can perform better than all baselines on improving semantic parsing. Finally, the conversational semantic parsing model trained by our method under semi-supervision can achieve 65% of the performance of full supervision.

Key words: Semantic Parsing, Weakly Supervised Semantic Parsing, Conversational Semantic Parsing, Compositional Generalization, Domain Generalization

目　录

图　目

表　目

第一章　绪论

1.1　背景与意义

在自然语言处理中，语义解析（Semantic Parisng）一直都是一个引人关注的话题。语义解析旨在将用户输入的自然语言解析成形式化的中间表示（Meaning Representation）。一般地，中间表示既可以是反映语义的形式化语义表征（Semantic Representation），也可以是面向任务可执行的计算机程序（Program）。本文关注在后者，即将自然语言解析成计算机程序，并在对应的执行器上执行完成复杂任务。值得注意的是，这里的程序并不局限在高级编程语言（如 Python 语言）所编写的程序，而是指所有结构化的、机器可理解并执行的机器语言。一个典型的语义解析流程如图 1所示。

图 1　语义解析流程示意图

语义解析适用的范围非常广泛，因为程序的种类有很多。一个语义解析系统的输出空间可以简单到只有几个 API（Application Programming Interface，应用编程接口），也可以复杂到是一门高级编程语言。例如，用于数据库查询的 SQL（Structured Query Language，结构化查询语言）语句，用于 Linux 系统操作的 Shell 命令，用于文本模式匹配的正则表达式都可以是语义解析系统输出的程序。在自然语言处理领域，语义解析一直都非常引人注目，它对于研究者和终端用户都非常有价值。

对于研究者来说，语义解析是在理解模糊的、带语境的、非结构化的人类语言道路上的一大里程碑，因此它具有重大的研究价值。纵观整个自然语言处理的发展历史，从信息检索（如常见的命名实体抽取，关注在一些预定义的关系和实体相关的信息），到语义分析（如摘要抽取，关注在文本浅层语义抽取和特定语言现象），再到语义解析（关注在文本的深层语义），任务的表达力是逐步上升的。从这样的角度来看，作为关注文本深层语义的语义解析技术，它已经可以向下兼容解决许多经典任务。例如，在任务型对话系统（Task Oriented Dialog）研究领域，已经有研究者将任务型对话视作层次化语义解析任务，并使用语义解析的方法来解决[1]。

对于用户来说，语义解析能够让用户无需学习任何程序，只需自然语言就可完成原本需要复杂的程序才能完成的任务，因此它具有很高的商业价值。时至今日，语义解析

系统已经可以生成 SQL 语句满足数据分析的需求[2]，生成 Shell 命令满足控制 Linux 操作系统的需求[3]，以及生成机器人的移动指令满足用户控制机器人的需求[4]。

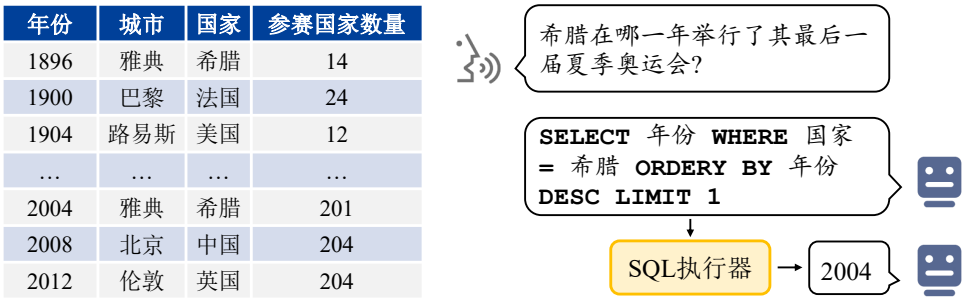

图 2　语义解析典型样例：text-to-SQL

语义解析系统是面向任务的，而任务往往有相应的知识库（如表格），因此语义解析中一支重要的工作就是知识库背景下的语义解析。和没有知识库背景的语义解析系统相比，带有知识库背景允许用户为模型提供自定义的输入，从而允许一个训练好的语义解析模型有能力泛化到不同的知识库上。图 2展示了基于知识库语义解析的典型样例，一个为给定的自然语言生成对应 SQL 语句（text-to-SQL）的语义解析系统。在该场景中，用户可以提供一个表格作为用户端的知识库，例如图 2中所展示的与奥运会项目有关的表格。如果用户想从冗长的表格中找出他关心的数据内容，他不需要学会数据库查询语言 SQL，而是可以直接用自然语言提问，比如这里的“希腊在哪一年举行了其最后一届夏季奥运会?”。一个 text-to-SQL 语义解析系统的目标就是生成该问句对应的 SQL 语句（如图中的 SELECT 年份 WHERE 国家 = 希腊 ORDERY BY 年份 DESC LIMIT 1），并且通过在 SQL 执行器（如 MySQL）上执行找出用户感兴趣的表格数据返回给他（如图中的 2004）。

知识库并不局限在结构化表格，也可以是知识图谱。如图 3所示，在基于知识图谱的语义解析方法加持下，谷歌搜索引擎已可以返回部分事实型问题对应的答案。类似地，语义解析可以支持的下游任务包括但不限于数据分析、问答系统、虚拟助手等。通过对这些复杂任务的支持，语义解析系统普遍提高了用户完成任务的效率，助力了许多知名商业产品的诞生，如 WolframAlpha 知识引擎[1]。如上所述，语义解析有着重大的研究价值和很高的商业价值，因此**本文的研究工作主要聚焦在语义解析研究方向**。

1.2　国内外研究现状

根据相关技术的研究背景，针对其中的核心关键问题，本文根据领域的特点，将从**程序驱动的强监督语义解析**，**答案驱动的弱监督语义解析**和**对话式语义解析**三方面展开

[1]https://www.wolframalpha.com/

图 3　语义解析在谷歌搜索引擎中的真实场景应用

本领域的调研与分析。

1.2.1　程序驱动的强监督语义解析

程序驱动的强监督语义解析是最经典的语义解析任务设定，本节主要从数据集和模型架构两方面介绍程序驱动的强监督语义解析的代表性工作。下文中，如无特意指出，**语义解析**一般指的即是无语境的、程序驱动的强监督语义解析。

相关数据集

早期的语义解析数据集都是**限定域**的，即指训练和测试时模型都在同样的领域（Domain）下，这种情况下小规模的数据集就足以支撑一个语义解析器的学习。1996 年，Zelle 等[5] 提出了 GEOQUERY 数据集，语义解析最经典的数据集之一。以一个小型的美国地理知识库为背景，该数据集一共收集了 880 个查询的自然语言和对应的 FunQL 程序。GEOQUERY 数据集是限定在地理领域上的，后来也陆续有研究者提出了其他限定域的语义解析数据集，其中比较典型的有限定在工作领域的 JOBS 数据集[6]，限定在航空订票领域的 ATIS（Air Travel Information System，航空旅行信息系统）数据集[7] 等。

表 1　OVERNIGHT 数据集中 7 个领域与对应的自然语言样例[8]

领域	自然语言样例
日历（Calendar）	Show me meetings after the weekly standup day 翻译：每周例行站会后有哪些会议要开
建筑（Blocks）	Select the brick that is to the furthest left 翻译：选择最左边的那块砖
住房（Housing）	Housing that is 800 square feet or bigger 翻译：*800* 平方英尺或更大的住房
餐厅（Restaurants）	What restaurant can you eat lunch outside at 翻译：什么餐馆可以在外面吃午餐
学术（Publications）	Who has co-authored articles with Efron 翻译：谁与埃弗隆一起合著过文章
社交（Social）	When did alice start attending brown university 翻译：爱丽丝什么时候开始上的布朗大学
篮球（Basketball）	How many fouls were played by Kobe Bryant in 2004 翻译：科比布莱恩特在 *2004* 年有多少次犯规

针对单个领域的数据集应用较为局限，因此后来的相关工作在构建数据集时开始考虑多个领域。2013 年，Cai 等[9] 提出了 FREE917 数据集，首次在训练语义解析模型时融合了多领域数据。FREE917 数据集一共包含 917 个查询的自然语言和它们对应的 λ-表达式，覆盖了 Freebase 知识库中的多个领域。但是该数据集中句子的结构不太复杂，大部分查询语句都仅包含一个实体，一个样例查询是 “What genre of music is B12?”（**翻译：B12 是什么类型的音乐**）。2015 年，Wang 等[8] 利用语法规则批量生成程序和对应的自然语言查询模板，再众包转述（Paraphrase）自然语言查询构建了一个涵盖 7 个不同领域的数据集 OVERNIGHT，各个领域与对应样例见表 1。对于样例程序如 type.housingUnit $\cap$ area. $>$ .800 来说，研究者们会首先自动生成一个语义正确但语法不流畅的自然语言如 “housing unit whose size is at least 800 square feet.”（**翻译：面积至少为 800 平方英尺的住房单元**），众包改写后的自然语言更加流利如 “Housing that is 800 square feet or bigger?”（**翻译：800 平方英尺或更大的住房**）。至此，OVERNIGHT 成为首个程序标注规模超过 10,000 的多领域语义解析数据集。

虽然数据集标注的数量已经小有规模，但 OVERNIGHT 和以前的语义解析数据集一样，仍然关注限定域场景下的语义解析。这样导致的一个缺陷在于，在这些数据集上训练的语义解析器很难泛化到新的领域中。例如，当训练数据是天气相关的领域，测试时语义解

Easy

What is the number of cars with more than 4 cylinders?

```
SELECT COUNT(*)
FROM cars_data
WHERE cylinders > 4
```

Meidum

For each stadium, how many concerts are there?

```
SELECT T2.name, COUNT(*)
FROM concert AS T1 JOIN stadium AS T2
ON T1.stadium_id = T2.stadium_id
GROUP BY T1.stadium_id
```

Hard

Which countries in Europe have at least 3 car manufacturers?

```
SELECT T1.country_name
FROM countries AS T1 JOIN continents
AS T2 ON T1.continent = T2.cont_id
JOIN car_makers AS T3 ON
T1.country_id = T3.country
WHERE T2.continent = 'Europe'
GROUP BY T1.country_name
HAVING COUNT(*) >= 3
```

Extra Hard

What is the average life expectancy in the countries where English is not the official language?

```
SELECT AVG(life_expectancy)
FROM country
WHERE name NOT IN
   (SELECT T1.name
    FROM country AS T1 JOIN
    country_language AS T2
    ON T1.code = T2.country_code
    WHERE T2.language = "English"
      AND T2.is_official = "T")
```

图 4　Spider 数据集四种难度示例[10]

析系统也只能完成天气查询的功能，而无法跨领域泛化（Cross-Domain Generalization），完成如查询会议室等功能。2017 年，来自华盛顿大学和 Salesforce 研究院的 Zhong 等[11]向跨领域泛化的设定迈进了一大步，提出了模型在训练时“不可见”（Unseen）测试时涉及表格的 WikiSQL 数据集。WikiSQL 数据集的构造借鉴了 Overnight 规则合成加人工转述的数据集收集方法。这种方法对标注的需求从编写程序降低到仅需转述规则生成的自然语言，也因此可以相对大规模地进行众包构造。最终，WikiSQL 收集了 87,673 个自然语言查询和对应的 SQL 语句，成为了至今最大规模的语义解析数据集。即使这种数据集收集方法较容易规模化，它并不能较好地还原真实用户的自然语言数据分布，也较难涵盖比较复杂的语义。对于 WikiSQL 来说也是如此，该数据集包含的程序复杂度较低，SQL 语句中不涉及到常见的排序（Order）、分组（Group）等操作，也没有嵌套查询等高级语法。2018 年，来自耶鲁大学的 Yu 等[10] 构造了首个跨域的、多表数据库上的 text-to-SQL 数据集 Spider。Spider 数据集涵盖了横跨 138 个领域的 200 个多表数据

库，一共包含超过 10,000 个自然语言查询与复杂 SQL 的成对数据。图 4展示了该数据集上四种不同难度的 SQL 样例，可以看出该数据集的多样性较好，包含了不同复杂程度的 SQL 语句。Spider 的出现掀起了研究者们对开发跨领域 text-to-SQL 系统的极大热情，时至今日已经有超过 70 个系统在 Spider 上进行过评测[2]，而该数据集也成为目前跨域语义解析最有挑战性的数据集之一。

图 5　HeartStone 数据集示例[12]

语义解析与代码生成领域也密切相关，其中的一个典型代表就是 text-to-Python。已知最早的相关工作是 2015 年提出的 Django 数据集[13]，该数据集先收集了较多 Python 程序，然后为它们标注类自然语言的伪代码（Pseudo Code）。2016 年，人工智能研究机构 DeepMind 发布了 HeartStone 数据集[12]。在该数据集的设定下，系统所接受的自然语言是一组关于炉石传说卡牌的描述，而语义解析系统的目标就是生成一串能够实现这个卡牌的 Python 代码，一个输入输出的样例如图 5所示。Django 和 HeartStone 数据集引起了研究者们的兴趣，但它们对于用户来讲实用性并不强，因此后来研究者们又构建了 CoNaLa 数据集[14]，用于训练模型以自动化地把简短的自然语言问题变成相应的简单 Python 代码实现。数据集中的问题和对应的 Python 代码源自一些知名的编程问答平台（如 StackoverFlow）上的问题和相关回答，贴合实际场景。最近，以预训练语言模型 GPT 系列闻名的 OpenAI 和 DeepMind 也分别发布 text-to-Python 的数据基准 HumanEval[15] 和 CodeContests[16]。这两个数据基准全都是由人类专家构造，每个样例中均包括一组单元测试（Unit Test）以评估模型生成的 Python 程序是否正确。

[2]https://yale-lily.github.io/spider

相关方法

语义解析是一个历史悠久的基础研究领域，已知最早的语义解析系统可以溯源到1964 年的 STUDENT 系统[17]。该系统是典型的**基于规则的语义解析**，它通过人工构造覆盖领域中所有可能的语言现象，为特定领域算术问题开发了一个语义解析系统。但考虑到该系统的鲁棒性极差，机器学习兴起后，研究者们更多地转向将机器学习和传统语言学理论结合起来的**基于组合文法的语义解析**。

基本成分	程序空间
Nations	nation
Borders	next_to_1
Publication date	publication_time
Who	author

图 6　基于组合文法的解析器词典示例

基于组合文法的语义解析方法一般会借助预先定义好的词典将句子中的基本成分翻译到程序空间，再利用一定程度上可泛化的组合规则组合各个翻译的成分，最终形成完整的程序。一般来讲，将句子中的基本成分翻译到程序空间的过程被称为语义落地（Semantic Grounding），该过程依赖于自定义的词典，一个典型的词典如图 6所示。在基于组合文法的语义解析方法中，最经典的工作莫过于利用组合范畴文法（Combinartory Catregorial Grammar，CCG）进行语义解析的相关工作[18-20]。组合范畴文法表示将程序的句法和语义分离开，第一步先通过枚举每个自然语言中可能的词汇候选学习合适的句法映射关系（即词典学习），第二部再通过预先定义好的组合文法进行语义解析。例如，对于动词 “border” 来说，它的句法形式可以表达为 (S\NP/NP)，其含义是这个动词还需要在右侧搭配一个宾语，在左侧搭配一个主语，而该句法对应的程序空间表示则为.y.border(y,x)，其中 y 和 x 分别表示宾语和主语对应的语义。基于组合范畴文法的语义解析与语言本身的组合性有很好的契合度，但因为其表示过于复杂，且要求自然语言每个单词的语义都需要反映在程序表示中，对程序的设计要求很高。因此，后来它逐渐被基于依存的组合语义（Dependency-based Compositional Semantics, DCS）文法所取代。基于依存的组合语义文法最早由 Liang 等[21] 提出，该文法类似于依存树，可以借助当时已经日渐成熟的依存分析技术[22] 一起对自然语言进行解析。和组合范畴文法不同，基于依存的组合语义主要由语义规则组成，其中包括一元实体（如 Seattle），实体相关的二元谓词（如 PlaceOfBirth），实体无关的二元谓词（如 Join）等。后来，λ-DCS 的出现[23] 强化了该方法的表达能力。

基于组合文法的语义解析方法很好地利用了语言本身的组合性，遵从的是自底向上（Bottom-up）地组合单词语义的架构。然而，因为这类方法往往需要大量预先定义复杂的词典和组合规则，对于程序文法的要求较高。后来，研究者们发现机器翻译与语义解析之间的异曲同工之妙，而伴随着数据驱动的机器翻译的发展，自顶向下（Top-down）**基于翻译的语义解析**逐渐在语义解析占据了统治地位。

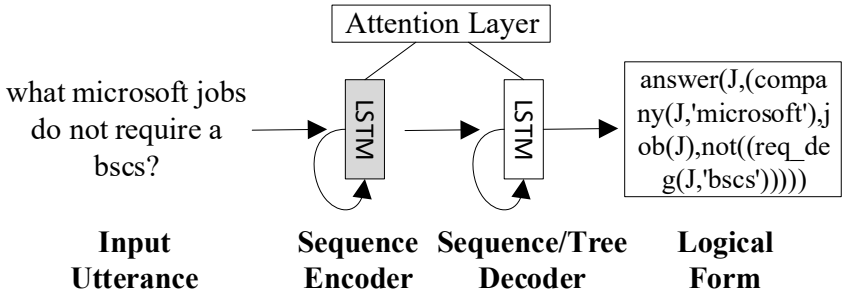

图 7　基于翻译的语义解析示意图[24]

基于翻译的语义解析方法的思想最早由 Andreas 等[25] 提出，但他们在实验中使用的仍是基于统计学习的机器翻译方法，如基于短语的翻译模型。在深度学习时代，基于编码器-解码器架构（Encoder-Decoder）的序列到序列模型（Sequence to Sequence）[26] 逐步成为机器翻译中的主流方法，Dong 等[24] 也在语义解析中使用带注意力的序列到序列（Sequence to Sequence with Attention）模型[27] 来建模。如图 7所示，相比于经典的组合文法方法，基于序列到序列模型的方法靠语义解析的标注数据驱动，不再需要手工定义词典和预先定义语义规则。

原始的序列到序列模型无法利用程序表示的结构信息，而程序表示所独有的结构信息是语义解析不同于机器翻译的重要特征。因此，后来的研究者们主要致力于如何让模型感知到程序表示的结构，即**基于程序结构的语义解析**。2018 年，Dong 等[28] 提出了由粗到细的解码方法（Coarse-to-Fine Decoding），将语义解析模型原先的一次解码变成两次解码。在第一次解码时，语义解析模型首先生成程序框架（Program Sketch），并利用该框架作为第二次解码的输入，生成最终的程序。以 SQL 语句为例，对于一个完整的 SQL 语句 SELECT Record Company WHERE (Year of Recording > 1996) AND (Conductor = Mikhail Snitko) 来说，其语义框架是 WHERE > AND =。可以看出，程序的语义框架是一种粗粒度的程序结构，通过这种先粗后细的解码方法，语义解析模型可以区分出程序的全局语义和局部语义。

程序的语义框架是一种结构信息，程序的语法则是另一种重要的结构信息。同样是 2018 年，多个研究工作先后提出了利用程序语法的语义解析系统[29-37]。这类语义解析方法将程序语法的先验融入语义解析器中，使得语义解析模型可以生成语法正确的程序。

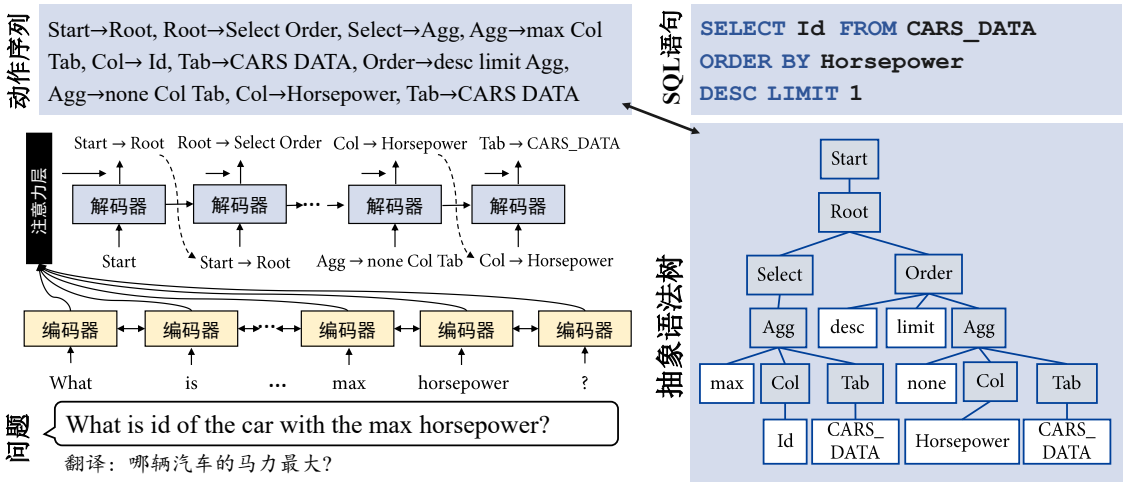

图 8　利用程序语法的语义解析示意图

与普通的基于翻译的语义解析模型将程序视为词的序列不同，利用程序语法的语义解析将一个程序视作一个动作序列。通过融入程序语法的知识，动作序列可以无损转换为一个语法正确的程序。以 SQL 语句为例，如图 8所示，与序列到序列模型中的编码器类似，利用程序语法的语义解析器中编码器也负责建模问句与知识库的表示，而解码器则根据编码器提供的表示，逐个选择有效的语法规则（Grammar Rule），最终得到一个动作序列（Action Sequence）。每个语法规则都包含一个非终结符（Nonterminal）和其展开得到的若干符号。在解码的每一步，都会有一个非终结符被一条其对应的语法规则展开，就形成了图 8中的动作序列 [Start→Root,...,Tab→CARS_DATA]。该动作序列可以通过从左至右深度遍历 SQL 语句 SELECT Id From CARS_DATA ORDER BY Horsepower DESC LIMIT 1 的抽象语法树（Abstract Syntax Tree）得到。

近些年，随着整个深度学习社区对模型泛化性的高度关注和相关数据集的出现，研究者们也开始关注语义解析模型的泛化性能，其中主流的两个研究方向分别是领域泛化（Domain Generalization）和组合泛化（Compositional Generalization）。领域泛化旨在提升模型的泛化性，使得它可以在从未见过的领域上性能同样优异。2019 年，来自微软亚洲研究院和西安交通大学的研究者们提出了 IRNet 模型[2]，首个在跨领域语义解析数据集 Spider 上 SQL 预测准确率超过 60% 的模型。研究者们认为现有的 text-to-SQL 模型的领域泛化能力较差，而其主要成因是大量域外词的存在。因此，他们通过启发式的方法将问题中的单词与知识库表格中的表名或列名链接在一起，再以链接后的问题作为语义解析模型的输入，以生成 SQL 语句。通过这种**实体链接**（Entity Linking）过程，IRNet 模型的领域泛化能力得到了显著的提升。2020 年，在该工作启发下，来自爱丁堡大学和微软研究院的 Wang 等[38] 提出了 RATSQL 模型，通过更精细地建模表格编码过程来进一步增强模型的领域泛化能力。具体地，RATSQL 模型中包含了一套关系感知的表格编码

方法（Schema-Aware Encoding)，这种编码方式在编码表格时会考虑到若干种关系，如表格名 cars_data 与自然语言中 “cars” 的链接关系，再如 cars_data 与其表内列名同表的关系等。通过将这些关系建模到 Transformer 模型的自注意力层中，RATSQL 能够更好地感知和表示关系，从而拥有更好的领域泛化能力。类似地，特拉维夫大学的研究者们利用图卷积神经网络[39] 强化了表格模式的表示，从而增强了语义解析模型的效果[40]。

近些年，预训练语言模型[41-43] 席卷了整个自然语言处理领域。通过在大规模的语料库上进行预训练，它们在多项下游任务上取得了可观的性能提升。因此，语义解析也出现不少预训练相关的工作用于增强模型的领域泛化能力，其中三个代表性工作分别是 TaPaS[44]、TaBERT[45] 和 GraPPa[46]。2020 年，来自谷歌的研究员提出了 TaPaS 模型[44]，这也是首个面向语义解析的预训练语言模型。TaPaS 从维基百科上爬取了大量表格和表格附近的自然语言段落作为模型的输入，直接利用 BERT[41] 中的掩码语言模型（Masked Language Model）目标继续预训练，在以表格为领域知识库的语义解析任务上取得了很好的效果。同一年，Yin 等[45] 提出了一个可以适用于表格理解的预训练语言模型 TaBERT[45]。在模型架构上，TaBERT 通过扩展 Transformer 模型中的自注意力（Self Attention）机制，让模型不仅可以注意到水平方向的单元格，也可以注意到垂直方向的单元格内容，从而实现高效编码表格内容的目的。在预训练时，TaBERT 的预训练语料库要更大，包含了将近 2,600 万表格和自然语言的成对数据，并采用类似掩码语言模型的目标，通过预测掩盖的表格列名（Masked Column Prediction）来实现预训练。该方法可以搭配不同的语义解析模型，统一地提升这些模型的领域泛化能力，再次证实了预训练对语义解析模型领域泛化能力的有效性。TaPaS 和 TaBERT 主要通过爬取大规模的相关数据完成预训练，而 GraPPa 另辟蹊径，利用下游数据集中的模板系统性地构造大规模的预训练数据。其构造数据的核心思想与前人工作类似[47]，研究者们首先通过现有的数据集人工归纳出同步上下文无关文法，接着利用这些文法系统性地合成新的自然语言和程序的成对数据。由于这些数据是合成的，因此对于每个自然语言 GraPPa 都是可以知道其对应的 SQL 语义。通过在原有的掩码语言模型的基础上引入 SQL 语义预测（SQL Semantic Prediction）的预训练目标，GraPPa 在多个语义解析任务上增强了已有模型的领域泛化能力。最近，也有不少工作将语义解析的输出程序转换成受控自然语言[48-49]，这样语义解析模型的输出更加接近语言模型在预训练时见过的样本，从而增强了语义解析模型的泛化性。

组合泛化是语义解析泛化研究工作中另外一个新兴方向，在该方向上研究者们主要关注在新的网络架构和数据增强的方法。在通过设计新的神经网络架构来提升语义解析模型的组合泛化能力的工作中，研究者们尝试了将句法和语义解耦[50-51]，设计神经符号

架构[52-54]，以及将问题建模成程序合成来解决[55]。在数据增强的方向上，Andreas[56] 通过采样真实的训练样本并随机替换其中的片段构造增强数据，Gordon 等[57] 通过预定义置换群构造增强数据使语义解析模型主动发现短语之间的等价性。最近，有研究者将元学习与组合泛化联系在一起[58-59]，也有研究者提出用统计翻译中的词对齐矩阵约束语义解析模型中的注意力分布以提升语义解析模型的组合泛化能力[60]。

1.2.2 答案驱动的弱监督语义解析

在前人工作中，语义解析器的训练都是需要程序标注的，但程序本身的复杂性注定其数据标注需要精通目标程序的专家，数据标注的门槛与成本都很高。考虑到语义解析器的目标仍然是完成下游任务，因此研究者们开始构建标注需求更弱的弱监督语义解析（Weakly-Supervised Semantic Parsing），从标注问题对应的程序变成了仅标注问题对应的答案。

相关数据集

2013 年，Berant 等[61] 提出了基于公开知识库 Freebase 的弱监督语义解析数据集 WebQuestions。虽然该数据集中的自然语言问句结构并不复杂，如 “What is the name of justin bieber brother”（**翻译：贾斯汀比伯兄弟的名字叫什么？**），但它是弱监督语义解析首个大规模的公开数据集，包含了多达 5,800 条数据。2015 年，Pasupat 等[62] 提出了 WikiTableQuestions 数据集，该数据集在维基百科上收集的超过 2,000 张半结构化表格上标注了超过 20,000 个复杂的自然语言查询和它们对应的答案。这些自然语言查询涉及了各种操作，如比较、极值和聚合计算。有些问题甚至很难通过 SQL 语句获得答案，例如一些问题的答案是某个单元格值的子串。虽然非常复杂，但因为 WikiTableQuestions 是用户自己完成的大规模标注，因此这些问题更符合用户的真实数据分布。同时，WikiTableQuestions 也首次将 “不可见” 表格的测试设定引入弱监督语义解析，即模型在测试和训练时所使用的表格是完全不同的，这也意味着模型必须要有能力泛化到新的表格上才能在该数据集上取得成功。因此，WikiTableQuestions 时至今日仍然是目前弱监督语义解析最有挑战的数据基准之一。

相关方法

因为弱监督的场景仅提供了答案，对语义解析模型来说学习更加困难。因此，研究者们在弱监督语义解析中尤其关注模型的学习算法，分别形成了**基于强化学习的算法**，**基于极大边缘似然的算法**和**可微分的算法**。

基于强化学习的算法的核心思想是首先让语义解析模型在输入自然语言后采样得到一组程序，然后通过对比这些程序执行的结果和标准答案得到奖励，再通过强化学习的算法优化语义解析模型以最大化奖励。这一类算法的代表工作有通过符号机器修剪程序搜索空间的 NSM（Neural Symbolic Machine，神经符号机）[63]，通过记录高回报的采样程序到记忆缓冲区（Memory Buffer）来减少强化学习训练时方差的 MAPO（Memory Augmented Policy Optimization，记忆增强的策略优化）[64]，通过构建辅助奖励函数为强化学习提供更精细反馈的 MeRL（Meta Reward Learning，元奖励学习）[65] 和通过为错误的采样程序生成对应自然语言加速强化学习收敛的方法[66]。

基于极大边缘似然的算法的核心思想是将程序视作隐变量，并通过结合启发式算法和标注的答案信息，将每个自然语言查询对应的程序空间从无穷大降低到一个可枚举的水平。在模型优化时，这一类算法直接优化在所有候选程序上的极大边缘似然（Maximum Marginal Likelihood）。在这一研究方向上，研究者们有使用类型约束降低程序搜索空间[62]，有通过将极大边缘似然算法中的系统搜索与强化学习的随机探索相结合使语义解析模型更容易输出一致程序[67] 和将期望最大化算法变成离散隐变量优化问题来解决的方法[68]。

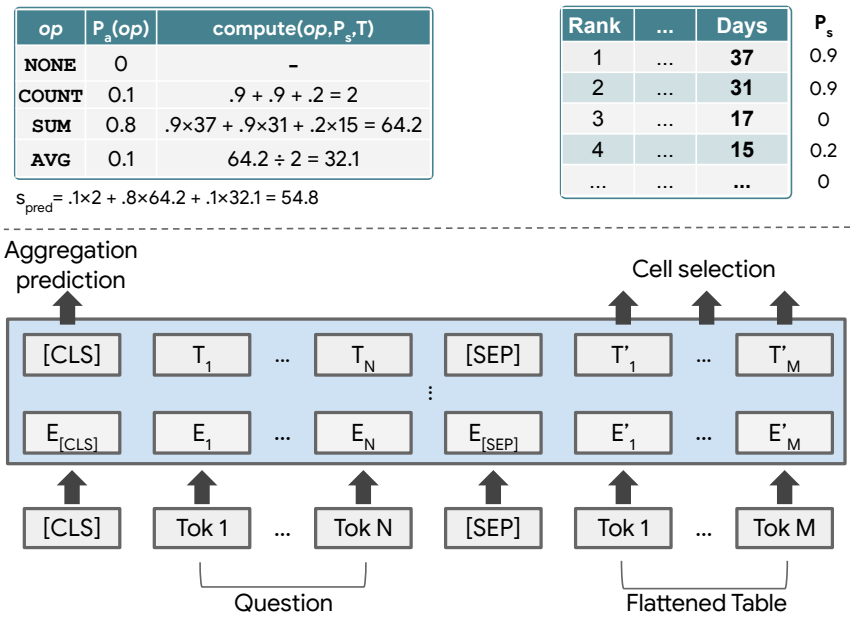

图 9　TaPaS 方法端到端地建模了弱监督语义解析任务[44]

基于完全可微的算法的核心思想是直接通过梯度下降来优化弱监督语义解析模型。2017 年，Neural Programmer（神经编程机）[69] 的提出首次为弱监督语义解析带来完全可微分的算法。在该方法中，研究者们通过在表格上执行固定次数的 “软” 操作（如取大于号）得到最终答案，从而使模型可以端到端地进行优化。2020 年，TaPaS 模型将多次操作变为一次，每次模型仅执行区域选择和区域上进行聚合运算两类操作（图 9），最

终在弱监督语义解析任务中取得了不错的效果[44]。

1.2.3 对话式语义解析

在语义解析，前人工作绝大多数集中在将一个语境无关的问题转换为其相应的程序。然而，真实的人机互动中用户往往会问一连串的问题，以了解一个特定的主题或实现一个复杂的目标。对话的形式对数据集的构建又提出了更高的挑战，因此到目前为止只有很少对话式语义解析（Conversational Semantic Parsing）的数据集。

相关数据集

show me flights from seattle to boston next monday
[Table with 31 flights]
on american airlines
[Table with 5 flights]
which ones arrive at 7pm
[No flights returned]
show me delta flights
[Table with 5 flights]
...

图 10 ATIS3 数据集对话示例[70]

对话式语义解析方面的工作，最早要追溯到 1994 年由 Deborah A. Dahl 等人所发布的 ATIS3 数据集[70]。ATIS3 数据集关注在航空领域的多轮对话问题，主要由美国和加拿大 46 个城市客服与用户的对话构成。这些对话涵盖了航空问答的常见场景，如查询所有可行航班，给定具体信息查询转机等，一个具体的对话示例如图 10所示。由于该任务主要是为用户展示航班信息数据，所以 ATIS3 数据集为用户所描述的自然语言标注了相应的 SQL 语句，这些 SQL 语句可以在航空系统的多表数据库中进行查询，得到相应的查询结果返回给用户。由于数据库的复杂性，ATIS3 数据集中的 SQL 语句都较为复杂，且跟语境紧密相关。例如，图 10中第一句话对应的 SQL 语句是 (SELECT DISTINCT flight.flight id FROM flight WHERE (flight.from airport IN (SELECT airport service.airport code FROM airport service WHERE airport service.city code IN (SELECT city.city code FROM city WHERE city.city name = ’ SEATTLE’))) AND (flight.to airport IN (SELECT airport service.airport code FROM airport service WHERE airport service.city code IN (SELECT city.city code FROM city WHERE city.city name = ’ BOSTON’))) AND (flight.flight days IN (SELECT days.days code FROM days WHERE days.day name IN (SELECT date day.day name FROM date day WHERE date day.year = 1993 AND date day.month number = 2 AND date day.day number = 8))))。

虽然 ATIS3 数据集中的 SQL 语句已经足以测试模型在生成复杂 SQL 方面的能力，

但由于它受限于航空领域，在它上面训练得到的模型很难泛化到其他领域，例如查询学校课程等。于是，研究者们开始构造更符合真实场景的对话式语义解析评测基准，也就衍生了跨域对话式语义解析（Cross-Domain Conversational Semantic Parsing）。2019 年，来自耶鲁大学的 Yu 等人贡献了两个数据规模大、涵盖领域多、难度较复杂的对话式语义解析数据集 SParC[71] 与 CoSQL[72]。从形式上说，SParC 与 ATIS3 数据集中的对话形式非常接近，都是由用户发出自然语言查询，系统显示通过 SQL 语句查询得到的结果。具体而言，SParC 数据集由 4,298 个人机对话和其对应的 SQL 语句组成，这些 SQL 语句关联到 138 个不同领域的 200 个多表数据库。然而，虽然同为跨域对话式语义解析，CoSQL 数据集与 SParC 却略有不同。在构造 CoSQL 数据集时，研究者们考虑到了更进一层的对话系统，即系统除了应该可以处理正常的自然语言查询外，也应该可以处理带歧义的问句。为应对 CoSQL 的场景，一个语义解析系统首先要能识别用户的查询是否有歧义，如有歧义则需要通过反问的方式让用户澄清自己的自然语言。

图 11　CHASE 数据集中对话示例[73]

SParC 和 CoSQL 数据集的主要动机是想让用户通过一组简单对话表达一个复杂意图，因此它在数据标注时也遵循一样的原则，即让标注人员将一个复杂 SQL 语句拆解，分散在对话的不同轮中表达。然而，这样的标注方法会带来一些问题，其中最主要的就是会造成 SQL 复杂度分布不均匀。例如，在 SParC 数据集中，第一轮问句对应的 SQL 往往非常简单，而最后一轮问句对应的 SQL 往往非常复杂，这与真实场景是不相符的。为了缓解数据集与真实情况下的分布差异问题，Guo 等[73] 改进了数据标注流程，并发布

了一个新的中文对话式语义解析数据集 CHASE。CHASE 中各轮 SQL 的复杂度比较均衡，它也鼓励标注员在对话中开启一个全新的话题，也因此更具挑战性。图 11展示了 CHASE 数据集中的一个示例，可以看出对话中自然语言的过渡更加流畅，数据集与真实世界分布更加靠近。

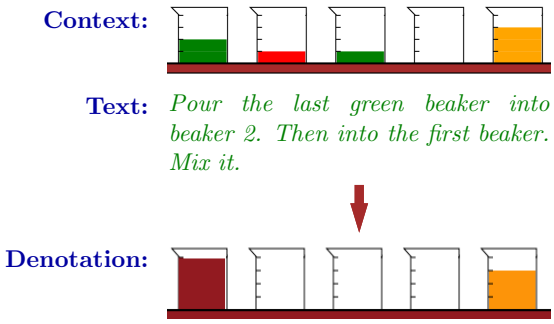

图 12　ALCHEMY 数据集示例[74]

在对话式语义解析的范畴，语境的定义并不局限于自然语言，也可以是其他语境。2016 年，Long 等[74] 首次提将语境的含义扩展到环境状态（World State）。图 12展示了研究者们所构建的 ALCHEMY 数据集的一个样例。在该数据集的定义中，烧杯与溶液的状态构成对话的背景，它们与历史对话一起构成了对话的语境。这就意味着，系统不仅需要理解一些代词如 “it” 在历史自然语言中所指代的实体，还要理解其在环境中所指的具体物体。

```
public class SimpleVector implements Serializable {
  double[] vecElements;
  double[] weights;

  NL Query: Adds a scalar to this vector in place.
  Code to be generated automatically:
  public void add(final double arg0) {
    for (int i = 0; i < vecElements.length; i++){
      vecElements[i] += arg0;
    }
  }

  NL Query: Increment this vector
  Code to be generated automatically:
  public void inc() {
    this.add(1);
  }
}
```

图 13　CONCODE 数据集示例[75]

2018 年，Iyer 等[75] 贡献了一个对话式语义解析数据集 CONCODE，该数据集要求一个语义解析系统不但能够理解自然语言的语境，同时还要能够理解对话所处的代码语境。图 13展示了 CONCODE 数据集中的一个具体示例，该示例展示了一个 Java 语言编

写的一个类，程序员用户试图通过描述 “Adds a scalar to this vector in place”（**翻译：把一个标量加在这个向量上**）让系统自动生成 add 方法。为了生成该段代码，模型需要理解 “this vector” 所指代的是 vecElements 变量，体现了代码语境必不可少的地位。

相关方法

因为对话式语义解析场景较为复杂和困难，早期的相关工作不多。1996 年，Miller 等[76] 开发了一个基于统计的语义解析系统，该解析器不仅可以处理语境无关的自然语言查询，也可以处理 ATIS3 数据集中的语境省略现象。例如，当用户在上文中提到 “我想从波士顿飞往北京”，系统回复后，用户会自然地省略并伴随提问 “哪些航班周二有票”。但是这种隐含的约束条件并不往往都需要继承的，因此研究者们通过标注数据训练了一个统计模型来识别应该从语境解析的程序所继承的约束条件，以此来解决 ATIS3 上的省略现象。2009 年，同样是在 ATIS 数据集上，研究者们将类似的思想与当时最先进的组合范畴文法结合在一起[77]，采用一个两阶段的方法来为语义依赖于语境的自然语言生成对应程序。在第一阶段时，方法使用组合范畴文法对每个自然语言都产生一个依赖于上下文的程序表示。在第二阶段时，方法通过每个句子所处的语境对第一阶段产生的程序表示进行一系列修改，得到每个句子的最终表示。

在深度学习时代，研究者们更多地侧重于使用神经网络来建模对话语境的含义，其中的典型工作包括**基于分层编码的方法**，**基于复制操作的方法**和**基于数据流的方法**。

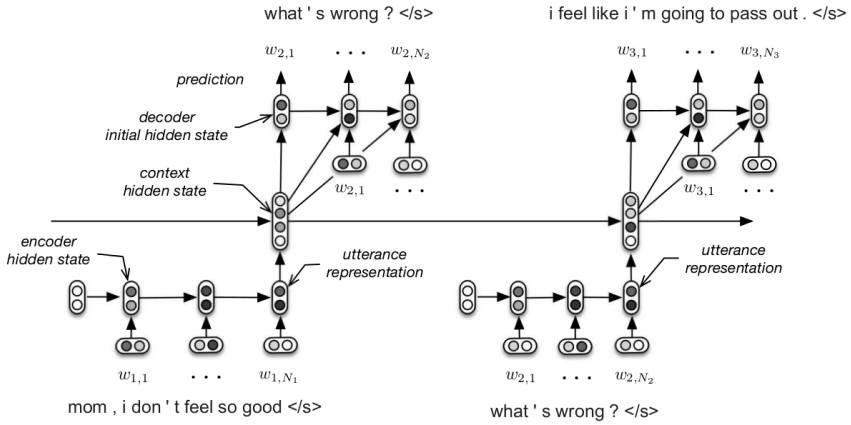

图 14　分层编码模型示意图[78]

2018 年，Suhr 等[79] 首次在对话式语义解析中应用基于分层编码的方法。该方法是由蒙特利尔大学的研究者们在 AAAI 会议上首次提出[78]，其核心思想是利用不同的编码器来捕捉不同粒度的信息流，从而从多个视角建模对话历史信息。如图 14所示，分层编码模型（Hierarchical Recurrent Encoder-Decoder，层次化的循环编码器-解码器）考虑到了多轮对话的两层结构，即一方面每个句子是由若干单词构成的，另一方面一个对话

又由若干句子构成。具体而言，该模型使用两个不同的 RNN（Recurrent Neural Network，循环神经网络）[80] 作为编码器来编码对话的层次结构。底层的 RNN 负责将每个自然语言句子映射为一个句子向量，这里的句子向量其实就是每个句子最后一个时间步由 RNN 输出的隐藏状态。高层的 RNN 则以每个句子向量作为输入，在模型内部将对话的所有历史信息整合在一起。而在 Suhr 等[79] 的论文中，研究者们发现分层编码不仅可以层次化地编码不同粒度的历史信息，也可以复用语境理解的计算量。相比于最朴素的将语境简单拼接送入编码器的做法，分层编码可以更高效地利用对话历史编码得到的隐藏状态。同时，为了让解码器能更好地区分不同轮次的编码状态，研究者们还为每轮的隐藏状态引入了相对位置信息。具体地，他们使用一个可学习的相对位置嵌入作为位置向量，位置向量与每一轮高层 RNN 输出的隐藏状态拼接在一起组成该轮的最终隐藏状态，解码器通过注意力机制将这些隐藏状态合并决策生成对话。

在对话式语义解析中基于复制操作的方法也是一支很重要的工作。Suhr 等[79] 提出通过复制充分利用对话中前几轮已生成的 SQL 语句，其主要动机来源于对话式语义解析中的一个普遍现象：对话中前后句对应的程序往往存在很大程度的重合。研究者们首先将 SQL 语句转换成 SQL 语法树，然后通过遍历得到了所有有效的子树对应的 SQL 片段（Segment）。传统的对话式语义解析器在解码的每一步都必须生成 SQL 语句的一个单词，而可复制的对话语义解析器的输出范围更大一些，其解码的每一步都要在 SQL 语句的词汇表和可复制的 SQL 片段中选择。在推理阶段，历史对话所生成的 SQL 片段都是可复制的候选片段。而在训练时，作者通过比较当前句与上一句标注 SQL 之间的重合部分，来确定当前 SQL 语句哪一部分是从上句 SQL 复制而来，剩下的部分则是通过依次生成 SQL 语句中的单词来构成的。

但实际上，这种方法也会带来不小的数据噪音，因为有些情况下对话中前后两句的 SQL 只是因为主题上的关联恰好有较多重复片段，而非真正的指代或省略现象。为了缓解这个问题，Zhang 等[81] 提出了一种新颖的基于单词级别的复制模型 EditSQL。EditSQL 的原理类似于 CopyNet[82]，核心思想在于，在解码生成当前句 SQL 的每一步时，都有一定概率从上一句 SQL 进行单词级别的复制。由于 EditSQL 模型无需人工导出训练数据，复制模式与生成模式的概率通过可以学习的门控单元混合在一起，也就避免了由人工导出数据带来的噪音。

除了可以对输入的历史信息和输出的程序进行建模外，最近来自微软的研究者们又提出了一种新颖的对话式语义解析的思路：用数据流合成（Dataflow Synthesis）[83] 来建模对话式语义解析任务。与传统的对话式语义解析为对话中的每个自然语言都标注完整的程序不同，研究者们提出要区分对待不同的自然语言查询。对于语境无关的自然语

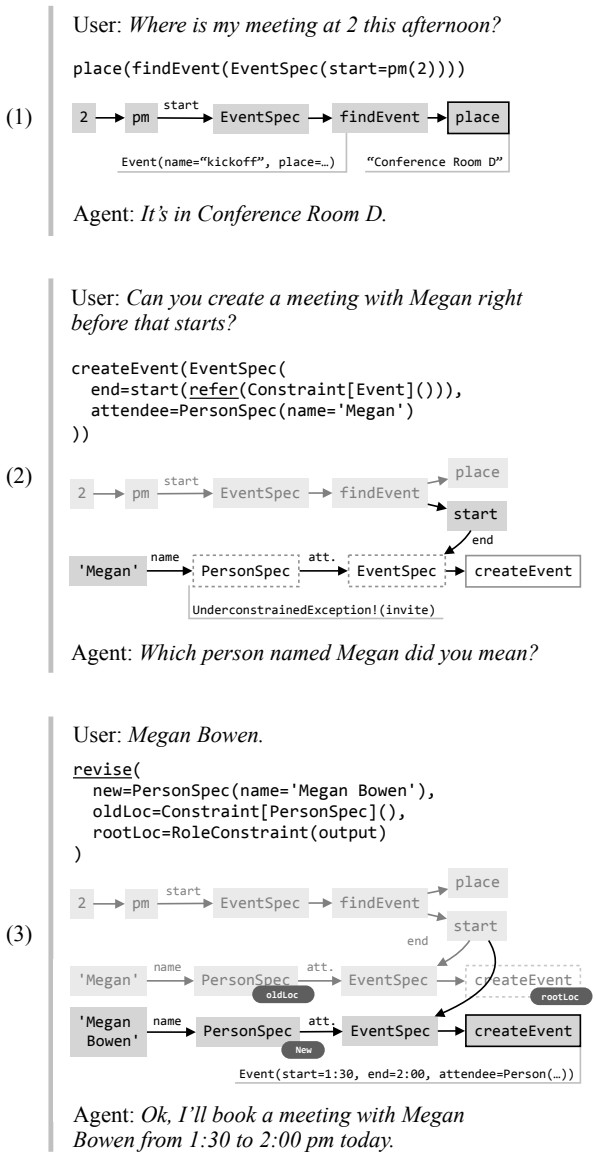

图 15　对话作为数据流方法示意图[83]

言，依然标注完整程序。但对于存在指代或省略的、语义依赖于语境的自然语言，研究者们提出使用两种元算子（Meta Computation），即指代（Reference）和修订（Revision），来标注这两种场景。指代主要用于对话中存在指代的现象，即当用某个代词指代上文中的某个物体时，比如“那个会议是几点开”。更新主要应对省略现象，一般用于用户希望更新历史状态时，比如“把会议改到 5 点”。这些元算子会用于维护一个随着对话动态变化的数据流，数据流可以转换成每个自然语言最终对应的程序，如图 15所示。

1.3 研究难点与挑战

近些年，随着深度学习技术的发展，语义解析有了很大的突破，并且已经被应用到了各种真实场景中。然而，语义解析仍存在着诸多挑战，其中业界公认的一大挑战在于**语义解析中程序标注的获取**。如上文所述，语义解析任务需要收集的数据中既包含自然语言文本，又包含与其对应的程序，而程序的复杂性注定其数据标注需要精通目标程序的专家，数据标注的门槛与成本很高。例如，Yu 等[10] 邀请了 11 名精通计算机专业的耶鲁大学本科生进行数据标注，总共耗费将近 1,000 个小时才收集好约 10,000 个训练样本的 text-to-SQL 数据集 Spider，数据标注非常耗时。而在 Yin 等[14] 的研究中，仅仅是核实与修订从 StackOverflow 上爬取的 text-to-Python 数据集 CoNaLa，每个程序都需要花费将近 1 美元。考虑到 CoNaLa 数据集中的 Python 程序都是形如 shutil.copy('file.txt', 'file2.txt') 的简单的单行程序，可以预见标注复杂程序的成本会显著更高，数据标注非常昂贵。

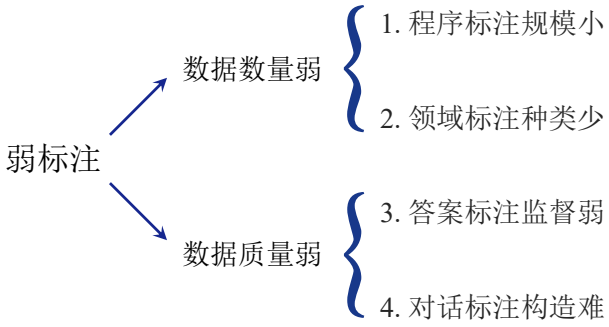

图 16　弱标注的四大难点

面对程序标注数据成本高的难题，研究者们提出开展**弱标注**下的自然语言语义解析研究，从而降低数据集的标注成本。弱标注意味着数据数量和质量两个维度的弱化，此时模型学习困难且容易过拟合，为语义解析方法的研发提出了极大挑战。如图 16所示，一方面，当方法能使用的数据数量较弱时，模型面临**程序标注规模小**和**领域标注种类少**的难题。如1.2.1节所述，目前基于翻译的语义解析方法在语义解析占据统治地位。该方法把编程语言看成一门外语，用类神经机器翻译的方法来解决语义解析，因此主流语义解析模型会将序列到序列模型作为基础[26]。然而，前人工作表明，序列到序列模型是数据饥饿的[84]。伴随着数据数量变弱，程序标注的规模和知识库对应领域的种类都相应变弱，语义解析模型非常容易过拟合到特定程序或特定领域上。另一方面，当方法能使用的数据质量较弱时，模型面临**答案标注监督弱**和**对话标注构造难**的难题。研究者们曾提出仅为自然语言标注答案来代替程序作为监督信号，答案标注相比程序标注众包标注的门槛和成本都显著降低。然而，弱质量意味着模型能从监督中获取的信息量少，前人的

工作也证实了语义解析模型在答案标注下往往无法取得与程序标注条件下一样好的性能[85]。类似地，语义解析研究者们的一大理想是构建可以理解人类对话的对话式语义解析模型，然而对话场景的标注成本很高，使用弱质量数据低成本地解决对话场景就成为扩展传统语义解析模型的殷切需求。然而，到目前为止，还没有研究者成功利用弱质量数据处理对话式语义解析任务。

1.3.1　难点 1. 程序标注规模小

标注自然语言对应的程序是非常昂贵的，因为只有精通相应程序的专家，因此研究者们一直在致力于降低对程序标注数据数量的需求。然而，由于程序是有组合性的，这意味着即使某种程序的语法非常简洁，有效的程序组合空间也很大。当程序标注的规模较小时，数据集所覆盖的程序组合数也会相应变少，模型从未见过的程序组合也就更多。若模型没有一定的组合泛化能力，它在真实场景中的性能会随着程序标注数量的减少而急剧下降。因此，**如何提升语义解析模型面向程序组合的泛化能力**是程序标注规模变小时语义解析模型所面临的主要挑战。

1.3.2　难点 2. 领域标注种类少

当前语义解析的研究者们主要关注在更符合真实世界设定的、跨域的语义解析模型上，因为语义解析模型在训练时不可能见过全部领域的知识库。然而，当数据标注数量较少时，语义解析数据集涉及到的领域种类也会相应变少。当训练样本覆盖的领域种类数量较少时，如果一个语义模型的领域泛化能力差，它就很难泛化到其他领域，导致语义解析模型可应用的场景非常局限。因此，**如何提升语义解析模型面向知识库的领域泛化能力**是领域标注种类少时语义解析所面临的主要挑战。

1.3.3　难点 3. 答案标注监督弱

与程序标注需要专家细致编写不同，普通人稍加思考就可以推理出自然语言问题对应的答案，因此研究者们开始使用答案监督替代程序监督以降低数据标注的成本。然而，答案标注的成本降低了，相应地模型学习的难度却变高了，因为自然语言理解是个复杂过程，而答案所蕴含的信息量一般都小于程序。前人的工作也证实了，当有程序标注时，语义解析模型一般可以获得比仅有答案标注时更好的性能[85]。因此，**如何提升弱质量答案标注下语义解析模型的性能**就成了弱监督语义解析所面临的主要挑战。

1.3.4 难点 4. 对话标注构造难

相比于每次都需要用完整的自然语言表达用户意图，多轮对话是更自然的人机交互窗口。然而，相比于单轮语义解析，对话标注数据的构造成本更高、难度更大，使用弱质量数据低成本地解决对话场景就成为扩展传统语义解析模型的殷切需求。到目前为止，还没有研究者成功利用弱质量数据解决对话式语义解析任务。因此，**如何设计合适的弱质量数据以驱动对话式语义解析场景**就成为了对话式语义解析所面临的主要挑战。

1.4 研究目标与研究内容

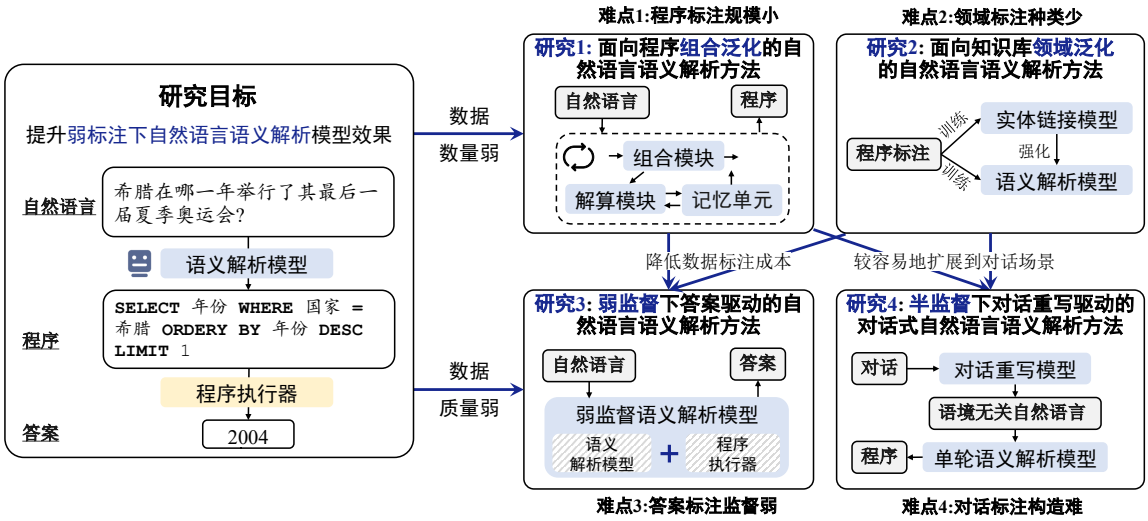

图 17　本文的主要研究内容及其关系

本文立足于自然语言语义解析，针对弱标注下的自然语言语义解析所面临的难点，计划从程序组合泛化性提升，知识库领域泛化性提升，弱监督下答案驱动语义解析模型性能提升，半监督下快速构建对话式语义解析模型等方面开展若干关键技术研究，最终完成**提升弱标注下自然语言语义解析模型的性能与泛化能力**的研究目标。研究目标与研究内容的关系如图 17所示，其中研究 1 与研究 2 面向的都是程序驱动的强监督语义解析场景，研究 3 面向的是弱监督语义解析场景，而研究 4 面向的是对话式语义解析场景。下文具体地解释了各个研究内容。

1.4.1 面向程序组合泛化的自然语言语义解析方法

现有的面向程序组合泛化的自然语言语义解析方法大部分都需要利用实验数据集的特点，期望通过增强数据弥补程序标注数量弱带来的性能损失。然而，由于这些人工规则是与数据集绑定的，无法普适于不同的语义解析数据集，因此这些方法较难扩展到

新的语义解析场景中。为了在不引入人工规则的情况下增强语义解析模型的程序组合泛化能力，本阶段重点研究新的网络架构来模拟人类的层次化抽象思维。本阶段拟设计多模块协同的网络架构，通过解耦的架构迭代理解自然语言，从而使模型拥有面向程序组合泛化的能力。

1.4.2　面向知识库领域泛化的自然语言语义解析方法

现有的面向知识库领域泛化的自然语言语义解析方法主要通过实现性能更好的实体链接模型将自然语言中提到知识库的信息剥离，从而使得语义解析模型更多地依赖领域无关的特征，实现语义解析模型的领域泛化能力。然而，已有的工作或者依赖于人工收集的高质量词典，或者需要人工标注实体链接的监督数据，较难规模化地扩展到各种语义解析场景中。为了能够规模化地增强语义解析模型的领域泛化能力，本阶段重点研究仅使用语义解析的数据学习实体链接模型，并强化语义解析模型。本阶段拟设计合适的中间任务从模型中导出实体链接的伪监督，并利用伪监督训练实体链接模型以强化语义解析模型面向知识库领域泛化的能力。

1.4.3　弱监督下答案驱动的自然语言语义解析方法

现有的答案驱动的自然语言语义解析方法可以分为基于强化学习、基于极大边缘似然和可微分的方法，然而这些方法都存在一定的缺点。基于强化学习或基于极大边缘似然的方法均需要预先使用启发式算法搜索自然语言对应的候选程序空间，而这些启发式算法都需要人工指定许多规则，繁琐的同时并不鲁棒。而去掉启发式算法，这些方法往往很难训练，模型不容易收敛。可微分方法的训练稳定性很好，因为它们可以通过反向传播直接优化模型参数。但已有的工作为了模型可以反向传播牺牲了模型的表达能力，从而无法建模较为复杂的程序（如 SQL 语句的嵌套查询）。为了在保持训练稳定性下同时保持较好的模型性能，本阶段重点研究更好的可微分方法来解决弱监督语义解析任务。本阶段拟设计表达能力更强的可微分方法和合适的预训练方法，从而让弱监督语义解析模型的性能与程序标注训练的模型性能可比。

1.4.4　半监督下对话重写驱动的对话式自然语言语义解析方法

现有的对话式自然语言语义解析方法主要通过从零开始众包标注对话，并为对话中的每个自然语言标注对应的程序。但这样的做法不仅标注成本昂贵，而且很容易因为标注方法不当导致数据分布不符合真实场景，从而使得数据集上训练好的对话式语义解析模型较难应对真实场景中的多轮对话。为了降低对话式语义解析方法的构建难度，本阶

段重点研究如何利用弱质量数据将已有的单轮语义解析模型迁移到对话式场景。本阶段拟采用对话重写作为弱监督信号来驱动对话式语义解析模型，并提出一个适合于对话重写的方法以提升语义解析模型在对话上的性能。

1.5 本文的组织结构

本文由六个章节组成，本章主要介绍研究背景与意义，国内外研究现状，以及研究的目标与内容，后续章节内容如下。

第二章介绍面向程序组合泛化的自然语言语义解析方法。本章针对程序标注规模小的难点，受启发于人类的层次化抽象思维，提出一个记忆单元增强的神经网络架构。由于该架构中模块之间的信息是离散的，该架构通过分层强化学习和课程学习的策略训练。最后，本章对所提的网络架构在不同的组合泛化挑战下进行了实验测试，验证了该模型的有效性。

第三章介绍面向知识库领域泛化的自然语言语义解析方法。本章针对领域标注种类少的难点，受启发于可解释机器学习相关研究，在预训练语言模型的基础上通过语义解析的程序标注数据训练得到一个实体链接模型，并通过该模型成功地增强了语义解析模型的领域泛化能力。最后，本章分别对所提出的实体链接模型和强化后的语义解析模型进行了实验测试，验证了所提出方法的有效性。

第四章介绍弱监督下答案驱动的自然语言语义解析方法。本章针对答案标注监督弱的难点，提出使用生成式模型同时发挥语义解析模型和程序执行器的作用，直接可微分地解决弱监督语义解析任务。在该方法的基础上，本章提出一个新的预训练方法，可以大幅度提升生成式模型在弱监督语义解析任务上的性能。最后，本章在三个知名的弱监督语义解析数据集上对所提方法进行了实验测试，验证了所提可微分方法解决弱监督语义解析任务的可行性与预训练方法的有效性。

第五章介绍半监督下对话重写驱动的对话式自然语言语义解析方法。本章针对对话标注构造难的难点，提出对话重写任务以复用已有的丰富的单轮对话语义解析的数据资源，并构建了一个新的对话重写数据集。同时，本章还提出一个基于拆分重组的对话重写方法，通过直接编辑对话更好地利用了对话本身的信息。最后，本章对所提的方法在所构建的数据集上进行了实验测试，验证了对话重写任务用于驱动半监督下对话式自然语言语义解析的可行性，且基于拆分重组的方法可以比所有基线模型更好地发挥语义解析模型的性能。

第六章对本文的研究工作进行总结，并对未来的研究方向进行展望。

第二章　面向程序组合泛化的自然语言语义解析方法

针对程序标注规模小的难点，为提升语义解析模型的泛化能力，本章提出一个记忆单元增强的神经网络架构，同时提出分层强化学习算法和课程学习的训练策略以稳定优化该架构，显著提升了语义解析模型面向程序组合泛化的能力。

2.1　引言

在理解自然语言时，人类有一种强大的泛化能力，能通过重新组合已知的短语来理解从未遇到过的新句子。例如，一旦人类学会了“鸭嘴兽”，“小蝌蚪”和“小蝌蚪在游泳”的意思，他们就可以轻松地理解“鸭嘴兽在游泳”的意思。这种能力依赖于语言所具有的组合性。组合性原则是指一个复杂的表达方式（如一个句子）的意义是由其成分（如名词“小蝌蚪”和动词“游泳”）的意义以及这些成分的组合方式（如一个名词和一个动词组成主谓结构）决定的。

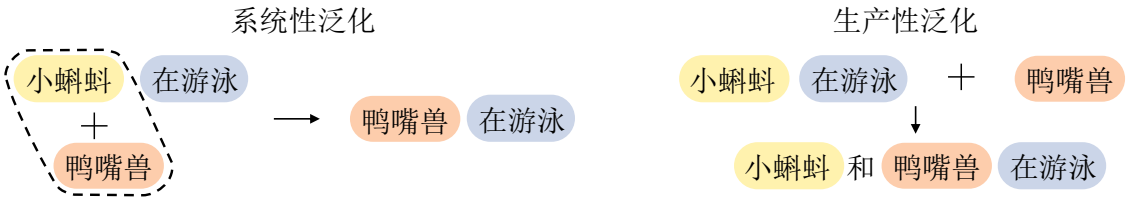

图 18　系统性泛化（左）和生产性泛化（右）的场景示例

从语言学的角度看，人类认知的组合泛化能力主要体现在系统性泛化（Systematic Generalization）和生产性泛化（Productive Generalization）。直观地，系统性泛化的能力表明人类可以理解已知语义的局部置换。如图 18左侧所示，当一个人理解了“鸭嘴兽”和“小蝌蚪在游泳”的含义，他可以自然地理解“鸭嘴兽在游泳”的含义。不同地，生产性泛化是指即使人类只学过有限的、相对简单的语义，但可以凭借潜在的普适规律，来理解无限的、相对复杂的语义。如图 18右侧所示，当一个人理解了“鸭嘴兽”和“小蝌蚪在游泳”这些相对简单的自然语句，那他一定能够理解复杂语句“小蝌蚪和鸭嘴兽在游泳”的含义。凭借着生产性泛化，人类在理论上可以理解无限长的自然语句[86]。正是依靠组合泛化能力，人类才能够从一些最基础的元素出发，一步一步创造出复杂甚至无限的语义世界。可以说，组合泛化是类人智能体必须具备的基本能力[87]。

相应地，语义解析模型也应当具有组合泛化能力，因为程序也是具有组合性的。任何一个训练数据集中所包含的程序，都只是巨大程序空间中的冰山一角。尤其是，当程

序标注规模小时，数据集所覆盖的程序组合数也会相应变少，模型从未见过的程序组合也就更多。若模型没有一定的组合泛化能力，它在真实场景中的性能会随着程序标注数量的减少而急剧下降。反之，语义解析模型如拥有面向程序的组合泛化能力，即使只有较小规模的程序标注，语义解析模型也有能力泛化地生成多样的程序组合。

图 19　SCAN 数据集: 从自然语言到导航动作序列[88]

从这个角度出发，越来越多的研究工作开始重新审视语义解析模型。2018 年，Lake 等[88] 的研究表明，主流的基于序列到序列架构的[26] 语义解析模型并不具有组合泛化能力。图 19展示了研究者们用于语义解析模型组合泛化能力的数据集 SCAN（Simplified version of the CommAI Navigation，简化的智能体导航），该数据集的任务是将如 “run twice” 的自然语言翻译成如 RUN RUN 的导航动作序列。如果按照深度学习社区经典的独立同分布设定，SCAN 是个非常简单的任务。例如，将 SCAN 随机地划分成训练集和测试集，再对模型进行训练，最简单的基于序列到序列语义解析模型也能在测试集上达到 99.8% 的准确率。然而，研究者们发现，一旦从组合性的角度对训练集和测试集的划分方式加以约束，基于序列到序列架构的语义解析模型就不再有效了。尽管模型看到过 “walk”, “walk twice” 和 “jump”，但语义解析模型很难将它们的语义推广并理解 “jump twice” 的含义。最近，一系列研究也表明，基于 Transformer 的、基于卷积序列架构等语义解析模型也不具备组合泛化能力[51, 57-58, 89-90]。

针对程序标注规模小的难题，为了让语义解析模型拥有面向程序组合泛化的能力，本章提出要将语义解析建模成一个多步迭代的过程，并使用记忆单元增强的神经网络架构来实现它。在人类的层次化抽象思维启发下，该网络架构被设计为一个记忆单元（Memory）与两个可学习的神经网络模块，即组合模块（Composer）和解算模块（Solver）。组合模块旨在从具体的自然语言中找到可抽象的局部自然语言片段，而解算模块则专注于通过访问记忆单元理解并映射这些片段到抽象的程序空间。这两个模块通过多步迭代，联合将自然语言逐步抽象，并通过合并每步产生的抽象程序生成最终的程序，实现语义解析。在 SCAN 数据集上的实验表明，本章提出的网络架构具有很强的组合泛化能力，在所有的任务中都达到了 100% 的准确率。同时，该架构也是第一个在没有额外人工规则的情况下通过 SCAN 上所有组合性挑战的神经网络模型。

2.2 总体结构

2.2.1 形式化定义

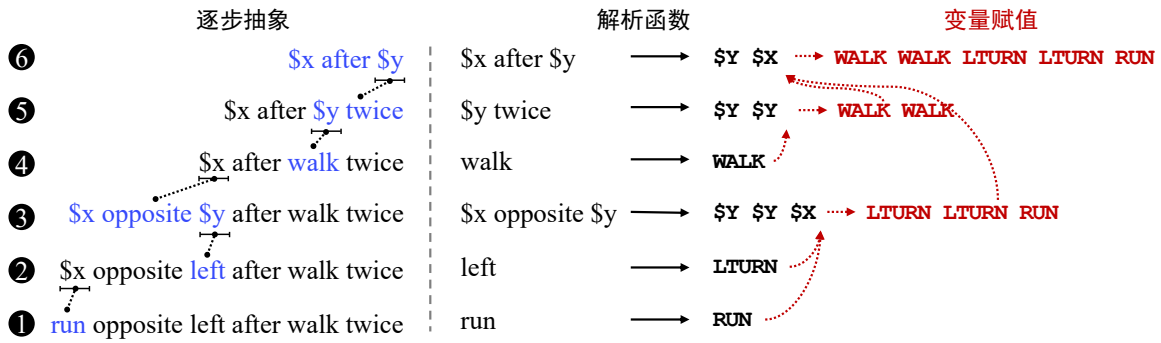

图 20　人类思维倾向于将具体对象层次化地抽象为具有潜在规律的抽象表达式

人类的认知之所以具备组合泛化能力，关键在于抽象（Abstraction）。抽象即省略事物的具体细节，以减少其中所含的信息量，从而更有利于发现事物间的共性规律。考虑这样一个实例，当见过 “run opposite walk”、“left after run”、“walk twice” 这些样例后，在理解新的语句 “run opposite left after walk twice” 时，人类会倾向于首先将该语句拆解成若干个训练数据中见过的句式，然后再逐个句式进行理解，最终将它们的语义合并起来解析为 WALK WALK LTURN LTURN RUN。人类的这种思考过程其实是一种层次化抽象的过程，认知科学家指出这种过程可以由带有符号函数（Symbolic Function）的变量槽（Variable Slot）来建模[86]。例如，任何附加有前缀 “super-” 的形容词都可以被视为在一个变量槽（如 “good”）上应用一个符号函数（即 “super-adj”），从而将语义映射到一个新的形容词（如 “super-good”）[86]。这样的建模可以将符号函数与特定的形容词解耦开来，使其能够泛化到全新的形容词（如 “bad”）上。受此启发，本章希望将符号函数和变量槽，即抽象表达式（Anaytical Expression）的归纳偏置引入语义解析模型中，以模拟人类的抽象化思维。

图 20左侧自底向上地展示了人类对于自然语言的层次化抽象化过程。在第 1、2、4 步，人类会屏蔽掉 “run”、“left” 和 “walk” 这三个词语的具体属性，并在下一步中将它们抽象为变量存在（如 “$x”）；在第 3、5、6 步，人类会屏蔽掉 “$x opposite $y”、“$y twice” 和 “$x after $y” 这三个子句的具体属性，也将它们抽象为变量。凭借这样的抽象思维，人类无需记忆复杂的从输入到输出的解析函数，而仅需要记忆抽象化过程每一步的相对简单的解析函数。例如，在第 1 步中，只需要记忆单词 “run” 的语义是 RUN；在第 3 步中，也只需要记忆抽象子句 “$x opposite $y” 对应的语义是 $Y $Y $X。此处的 $X 是一个程序中的变量，指代自然语言中的变量 “$x” 所对应的程序；同样地，$Y 亦是一

个程序中的变量，指代自然语言中的变量 “$y” 所对应的程序。从上述的示例可以看出，相比于直接记住相对复杂的具体映射，人类更倾向于从中归纳出相对简单的共性抽象映射，从而拥有组合泛化能力。

因此，为了让语义解析模型获得组合泛化能力，需要设计一种能够模拟人类层次化抽象的新型神经网络架构。具体地，本章提出了一个新型的神经网络架构 LANE（Learning Analytical Expressions）。它能够在语义解析任务中模拟人类的抽象化思维，从而获得组合泛化能力。在前人的语义解析框架中[26]，模型直接被用来学习一个直接从自然语言空间到程序空间的解析函数。而 LANE 既要学习将自然语言抽象化，又要学习从抽象自然语言空间到抽象程序空间的解析函数。由于 LANE 的输入输出有可能是抽象的，因此下文中分别称它们为语言表达式（Language Expression, LangExp）与程序表达式（Program Expression, ProExp）。若表达式中存在子部分被替换为变量，则称这些带变量的表达式为抽象表达式（Analytical Expression, AE），否则称其为常量表达式（Constant Expression, CE）。同样地，抽象表达式也可分为语言抽象表达式（Language Analytical Expression, LangAE）和程序抽象表达式（Program Analytical Expression, ProAE），常量表达式同样可以分为语言常量表达式（Language Constant Expression, LangCE）和程序常量表达式（Program Constant Expression, ProCE）。对于每个输入的 LangExp，LANE 需要通过若干次抽象化操作逐渐地将其转换为更简单的 LangAE（如图 20左侧所示）。在这一抽象化过程中，每一步都会有一个语言局部表达式（Language Local Expression, LangLE）被解析为一个程序局部表达式（Program Local Expression，ProLE）。通过变量赋值过程，每个 ProLE 都可以构造出一个 ProCE，最后一步的 ProCE 即是模型的输出（如图 20右侧所示）。模型需要以这种抽象化过程作为一种归纳偏置，在不依赖任何人工预定义的抽象与映射数据的前提下，自动化地完成对语言的抽象化过程并映射它们到程序空间。

2.2.2　方法概述

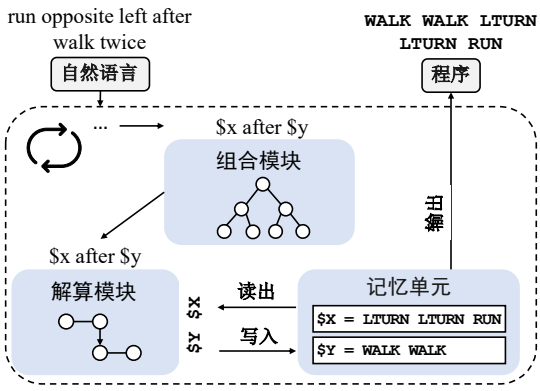

图 21　本章方法示意图：组合模块、解算模块与记忆单元迭代获得最终程序

如图 21所示，LANE 包含了三个部分：

组合模块 对输入的 LangExp（图中的 “$x after $y”）进行隐式树状归纳（Latent Tree Induction），从而得到 LangLE（图中的 “$x after $y”），实现图 20中的逐步抽象。

解算模块 以组合模块输出的 LangLE 作为输入，使用一个带注意力的序列到序列模型将其翻译为 ProLE（图中的 $Y $Y），实现图 20中的解析函数。

记忆单元 存储逐步抽象过程中变量对应的实际值，并据此将每个 ProLE 转换成 ProCE（图中的 WALK WALK LTURN LTURN RUN），实现图 20中的变量赋值。

为了理解一个自然语句，这三个模块缺一不可，模型需要它们迭代多步合作推理。自然语言是第一步时组合模块的输入，而记忆单元在最后一步所输出的 ProCE 即是整个模型的输出。

2.2.3 模型结构

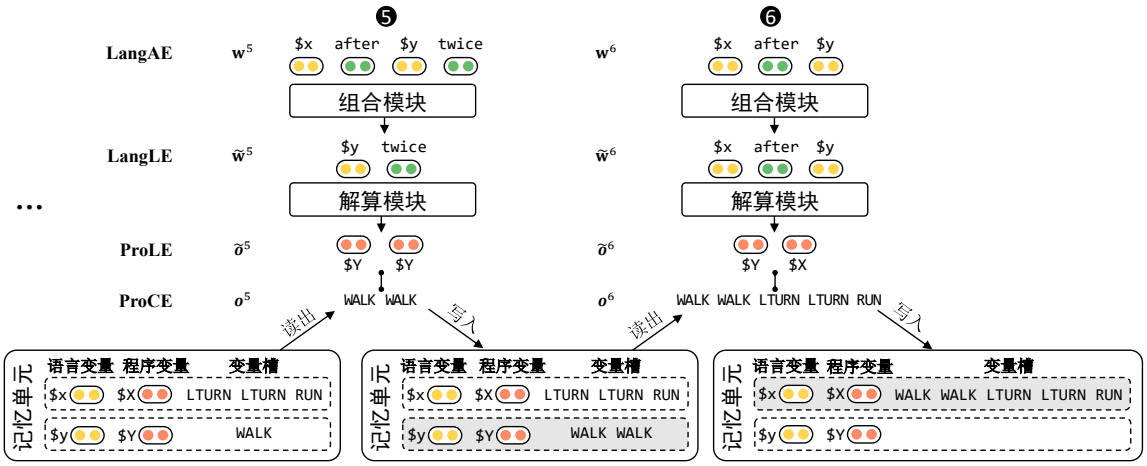

图 22 LANE 模型结构: 用神经网络学习隐式的抽象表达式

以图 20第 5 步和第 6 步的过程为例，图 22展示了 LANE 的模型结构。值得注意的是，图 22中彩色的神经元代表可学习的向量，其中绿色神经元代表动作词（如 “twice”）对应的词嵌入，黄色神经元代表语言变量（如 “$x”）的表征，红色神经元代表程序变量（如 $Y）的表征。在第 5 步时，“$x after $y twice” 被输入到组合模块中，组合模块决策要对 “$y twice” 进行抽象化，并将其送入解算模块中。以 “$y twice” 为输入，解算模块将 “$y twice” 翻译为 $Y $Y，同时与记忆单元中存储的变量槽 $Y = WALK 结合即到新变量所对应的 ProCE，WALK WALK，并据此更新记忆单元。经过这一过程，“$x after $y twice” 中的 “$y twice” 从原语义中被剥离，形成了一个抽象程度更高的 LangExp，即 “$x after $y”，进而开始第 6 步抽象。通过这样逐步抽象的方式，在组合模块、解算模块和记忆单元协同工作下，LANE 将一个自然语言句逐渐抽象成越来越短的 LangExp，直

到形成一个由单个变量构成的最小 LangExp。该变量在记忆单元中对应的取值即为最终输出的 ProExp。

2.3　组合模块

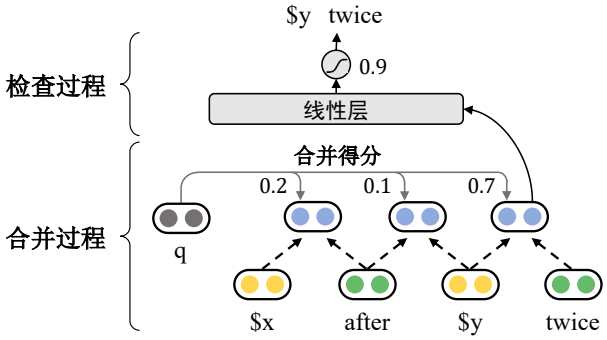

图 23　组合模块通过合并过程和检查过程寻找 LangLE

在第 t 步，以 LangExp $\mathbf{w}^t$ 为输入，组合模块旨在找到其中合理的 LangLE $\tilde{\mathbf{w}}^t$ 作为输出。为了实现该算法，LANE 采用了一种自底向上的方法，即尝试合并 $\mathbf{w}^t$ 中的每一对相邻单词，直到找到一个合理的 LangLE。如图 23所示，给定形如 “$x after $y twice” 的输入，组合模块首先会尝试合并 “$y” 和 “twice”。接着，它会检查 “$y twice” 是否可以构成一个合理的 LangLE。图示中的短语可以通过检查，此时组合模块会唤醒解算模块来解析该短语的语义。反之，如果 “$y twice” 无法通过检查，上述合并过程会迭代，即意味着组合模块会继续合并它和相邻单词，直到有一个合理的 LangLE 出现。直观地，上述过程可以视作自底向上地构建一棵二叉树，在每一层中，组合模块都会尝试合并两个相邻的节点并生长出它们的父节点（对应于合并过程），并检查该父节点是否对应于一个合理的 LangLE（对应于检查过程）。

2.3.1　合并过程

在合并过程中，组合模块首先枚举当前层所有可能的父节点并计算它们的表示，然后选择其中一个向上生长。假定第 $l-1$ 层中的第 i 与第 $i+1$ 个节点分别用 $\mathbf{r}_i^{l-1}$ 与 $\mathbf{r}_{i+1}^{l-1}$ 来表示，它们对应的父节点的表示 $\mathbf{r}_i^l$ 可通过 Tree-LSTM[91] 编码得到。与 LSTM[92] 的结构相似，Tree-LSTM 也是使用门控单元来控制从子节点到父节点的信息流。同时，对于每个节点而言，Tree-LSTM 都维护了一个隐藏状态（Hidden State）和细胞状态（Cell State）。这即意味着，$\mathbf{r}_i^l$ 是由一个隐藏状态向量 $\mathbf{h}_i^l$ 和一个细胞状态向量 $\mathbf{c}_i^l$ 构成的。对于任意父节点，它的表示 $\mathbf{r}_i^l$ 是通过在左孩子结点的表示 $\mathbf{r}_i^{l-1}=(\mathbf{h}_i^{l-1},\mathbf{c}_i^{l-1})$ 和右孩子结点的

表示 $\mathbf{r}_{i+1}^{l-1} = (\mathbf{h}_{i+1}^{l-1}, \mathbf{c}_{i+1}^{l-1})$ 上进行如下计算所得到的：

$$\begin{aligned}\begin{bmatrix}\mathbf{o}\\ \mathbf{f}_i^{l-1}\\ \mathbf{f}_{i+1}^{l-1}\\ \mathbf{e}\\ \mathbf{g}\end{bmatrix} &= \begin{bmatrix}\sigma\\ \sigma\\ \sigma\\ \sigma\\ \tanh\end{bmatrix}\left(\mathbf{W}_{\text{tree}}\begin{bmatrix}\mathbf{h}_i^{l-1}\\ \mathbf{h}_{i+1}^{l-1}\end{bmatrix} + \mathbf{b}_{\text{tree}}\right),\\ \mathbf{c}_i^l &= \mathbf{f}_i^{l-1}\odot\mathbf{c}_i^{l-1} + \mathbf{f}_{i+1}^{l-1}\odot\mathbf{c}_{i+1}^{l-1} + \mathbf{e}\odot\mathbf{g},\\ \mathbf{h}_i^l &= \mathbf{o}\odot\tanh\left(\mathbf{c}_i^l\right)\end{aligned} \tag{2.1}$$

式中 $\mathbf{W}_{\text{tree}} \in \mathbb{R}^{5D_h \times 2D_h}$ 是一个可学习的矩阵，$\mathbf{b}_{\text{tree}} \in \mathbb{R}^{5D_h}$ 是一个可学习的向量，σ 和 tanh 都是非线性激活函数，以及 $\odot$ 代表逐点乘积。至于叶子节点，它们的表示 $\mathbf{r}_i^l$（$l = 1$）可通过在它们相对应元素（如 “$x”）的表示上叠加线性变换得到：

$$\mathbf{r}_i^1 = \begin{bmatrix}\mathbf{h}_i^1\\ \mathbf{c}_i^1\end{bmatrix} = \mathbf{W}_{\text{leaf}}\,\text{Emb}\,(\mathbf{w}_i^t) + \mathbf{b}_{\text{leaf}}, \tag{2.2}$$

式中 $\mathbf{W}_{\text{leaf}} \in \mathbb{R}^{2D_h \times D_h}$ 是一个可学习的矩阵，$\mathbf{b}_{\text{leaf}} \in \mathbb{R}^{2D_h}$ 是一个可学习的向量，$\mathbf{w}_i^t$ 是 $\mathbf{w}^t$ 中的第 i 个元素。值得注意的是，当 $\mathbf{w}_i^t$ 为词语时，$\text{Emb}(\mathbf{w}_i^t) \in \mathbb{R}^{D_h}$ 代表其对应的词向量，而当 $\mathbf{w}_i^t$ 为语言变量时，$\text{Emb}(\mathbf{w}_i^t) \in \mathbb{R}^{D_h}$ 代表其在记忆单元中存储的语言变量表征。

在得到候选父节点的表示后，组合模块将依据**合并得分**选择应当向上生长的结点。如图 23所示，在得到所有候选父节点表示后（图中的蓝色神经元），组合模块先计算每个候选父节点的合并得分（图 23中的数字），再选择分数最大的父节点进行构建（带箭头的实线）。其中，合并得分是通过一个可学习的查询向量 $\mathbf{q}$ 与每个候选父节点表示 $\mathbf{r}_i^l$ 计算内积后经过 Softmax 函数得到的，它可用于衡量每个候选父节点构建的优先级。一旦第 l 层所构建的节点确定，组合模块就会进入到检查过程。

2.3.2 检查过程

检查过程旨在于检查合并过程所确定的新中间节点是否构成一个合理的 LangLE。具体而言，假定 $\mathbf{r}_i^l$ 为父节点表示，该节点通过检查的概率 p_c 可以通过如下公式计算:

$$p_c = \sigma(\mathbf{W}_{\text{c}}\mathbf{r}_i^l + b_{\text{c}}), \tag{2.3}$$

式中 $\mathbf{W}_c$ 与 b_c 都是可学习的参数（图 23中的线性层）。当 $p_c > 0.5$ 时，组合模块会唤醒解算模块对当前节点对应的短语进行翻译。否则，当前节点将与其他节点继续合并，从而进入新的一层，重新开始上述过程。

2.4　解算模块

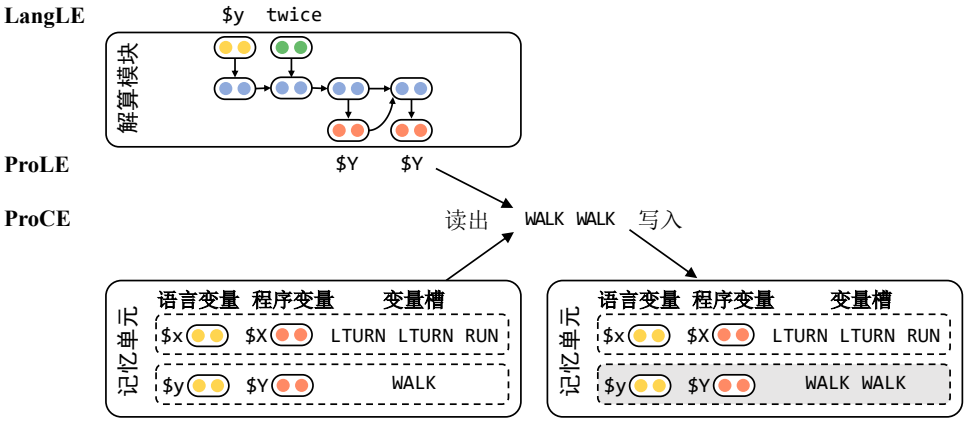

图 24　解算模块逐步解码生成 ProLE，并通过与记忆单元的交互产生 ProCE

2.4.1　生成程序框架

从组合模块处输出的 LangLE $\tilde{\mathbf{w}}^t$ 会送入解算模块，而解算模块的目标即将该 LangLE 翻译成对应的 ProLE $\tilde{\mathbf{o}}^t$，ProLE 可以视作程序的框架。为了实现这一目标，解算模块使用了带注意力机制的序列到序列网络[27]，该网络通过逐步解码产生 ProLE。在每一步解码时，解算模块会学习产生一个动作词（如 WALK），或产生一个程序变量（如 $X）。以图 24为例，解算模块以 “$y twice” 作为输入，输出一个带程序变量的程序抽象表达式，即 $Y $Y。

2.4.2　读写记忆单元

当解码过程结束后，解算模块会以 ProLE 为骨架，通过读取记忆单元中变量对应的实际值填充 ProLE 中的变量槽，从而得到一个新的 ProCE。同样以图 24为例，图中的 ProCE 是 WALK WALK。该 ProCE 将会以类似于前序步骤的方式，被重新写入记忆单元中，从而支持组合模块与解算模块的迭代求解。

2.5　模型训练

如果数据集中包含抽象表达式相关的标注（如 “$y twice” 和 $Y $Y 的成对数据），LANE 中的组合模块和解算模块就可遵循标准的监督学习（Supervised Learning）范式

进行学习。然而，由于数据集中并无相关数据，且额外标注抽象表达式又需要较高代价，LANE 无法通过监督学习分别学习这两个模块。又考虑到模块之间传递的信息（即 LangLE）是无法完全枚举的离散变量，LANE 也很难通过反向传播联合优化这两个模块。因此，LANE 应用强化学习算法完成整个模型的训练。

同时，从图 22中易看出，组合模块处于解算模块的上游地位，因为解算模块的输入需要由组合模块提供。受此启发，LANE 通过分层强化学习算法（Hierarchical Reinforcement Learning，HRL）[93-94] 的思想来建模两个模块的联合训练：组合模块是高层代理（High-Level Agent），负责寻找 LangLE；解算模块是底层代理（Low-Level Agent），在高层代理提供输入后输出相对应的 ProLE，最终通过与记忆单元交互产生 ProCE。

2.5.1 学习算法

作为强化学习算法中的一支，LANE 所使用的分层强化学习算法也包含了强化学习的两大基本要素：**状态**（State）和**动作**（Action）。由于 LANE 需要通过迭代多步建模问题，LANE 中的状态和动作都是与步数相关的。假定用 $\mathbf{s}^t$ 代表第 t 步状态，它不仅包括组合模块的输入，即 LangExp $\mathbf{w}^t$，也包括了记忆单元。对于组合模块，它在第 t 步的动作 $\mathcal{G}^t$ 就是它在第 t 步所找到的 LangLE。从环境中观测到 $\mathbf{s}^t$ 后，组合模块的参数 θ 定义了一个可学习的代理 $\pi_\theta(\mathcal{G}^t \mid \mathbf{s}^t)$。对于每个观测，$\pi_\theta$ 都会对应产生一个动作 $\mathcal{G}^t$，而该动作和状态 $\mathbf{s}^t$ 一起又会输入到解算模块中。类似于组合模块，解算模块的参数 φ 也定义了一个可学习的代理 $\pi_\varphi(\mathbf{a}^t \mid \mathcal{G}^t, \mathbf{s}^t)$，其中 $\mathbf{a}^t$ 是解算模块的动作，即每一步的 ProCE $\mathbf{o}^t$。从模型架构的角度来看，组合模块和解算模块地位是对称的，它们通过 $\mathcal{G}^t$ 耦合在一起。但从信息流的角度来看，组合模块和解算模块的地位是不对称的，因为解算模块要依赖于组合模块的输出。将组合模块视作高层代理，解算模块可以被视为底层代理，而 $\mathcal{G}^t$ 可以被视为组合模块给解算模块制定的子目标。

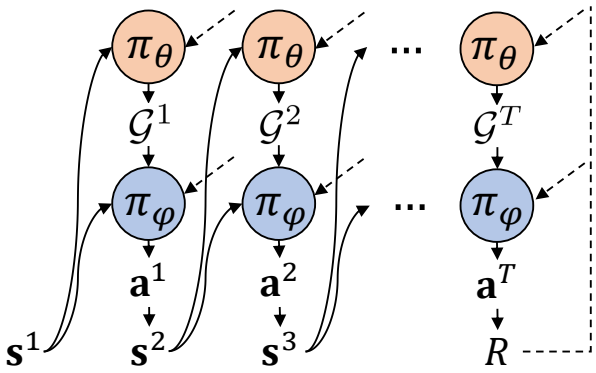

图 25　分层强化学习算法示意图: 红色的高层代理是组合模块，蓝色的底层代理是解算模块

(1) 策略梯度

图 25展示了 LANE 所应用的分层强化学习算法，可以看出组合模块和解算模块是交替产生动作的。第 t 步时，组合模块在参数 π_θ 的指导下选择一个高级动作 $\mathcal{G}^t$，并触发解算模块行动。类似地，解算模块在参数 π_φ 的指导下选择一个低级动作 $\mathbf{a}^t$。这两个模块交替行动，直到抵达终点（即第 T 步）并预测出最终的动作序列，并形成一条采样动作序列 τ 如下:

$$\tau = (\mathbf{s}^1\mathcal{G}^1\mathbf{a}^1 \cdots \mathbf{s}^T\mathcal{G}^T\mathbf{a}^T). \tag{2.4}$$

通过计算 τ 与真实的导航动作序列之间的重叠度，LANE 可以得到本次采样的**奖励**（Reward）。再通过策略梯度算法（Policy Gradient）[95]，收集到的奖励值可用于优化 θ 与 φ 参数，完成模型训练。具体地，假设 $R(\tau)$ 是 τ 对应的奖励（奖励函数将在2.5.1节中说明），LANE 的训练目标就是最大化奖励的期望：

$$\max_{\theta,\varphi} \mathcal{J}(\theta,\varphi) = \max_{\theta,\varphi} \mathbb{E}_{\tau\sim\pi_{\theta,\varphi}}\, R(\tau). \tag{2.5}$$

通过应用似然比技巧（Likelihood Ratio Trick），θ 与 φ 参数的优化可以从最大化公式 (2.5)中的目标函数，转变为循着下式中的梯度上升：

$$\nabla_{\theta,\varphi}\mathcal{J}(\theta,\varphi) = \mathbb{E}_{\tau\sim\pi_{\theta,\varphi}}\, R(\tau)\nabla_{\theta,\varphi}\log\pi_{\theta,\varphi}(\tau), \tag{2.6}$$

其中 $\pi_{\theta,\varphi}$ 可以进一步分解为一系列动作和状态的转移：

$$\begin{aligned}\pi_{\theta,\varphi}(\tau) &= p(\mathbf{s}^1\mathcal{G}^1\mathbf{a}^1 \cdots \mathbf{s}^T\mathcal{G}^T\mathbf{a}^T) \\ &= p(\mathbf{s}^1)\prod_{t=1}^{T}\pi_{\theta,\varphi}(\mathbf{a}^t,\mathcal{G}^t\,|\,\mathbf{s}^t)\,p(\mathbf{s}^{t+1}\,|\,\mathbf{s}^t,\mathcal{G}^t,\mathbf{a}^t).\end{aligned} \tag{2.7}$$

考虑到低级动作 $\mathbf{a}^t$ 的产生是以高级动作 $\mathcal{G}^t$ 为条件的，这就意味着以下等式的成立：

$$\pi_{\theta,\varphi}(\mathbf{a}^t,\mathcal{G}^t\,|\,\mathbf{s}^t) = \pi_\theta(\mathcal{G}^t\,|\,\mathbf{s}^t)\pi_\varphi(\mathbf{a}^t\,|\,\mathcal{G}^t,\mathbf{s}^t). \tag{2.8}$$

再结合链式法则，公式 (2.7)可以进一步被展开成为：

$$\pi_{\theta,\varphi}(\tau) = p(\mathbf{s}^1)\prod_{t=1}^{T}\pi_\theta(\mathcal{G}^t|\mathbf{s}^t)\pi_\varphi(\mathbf{a}^t|\mathcal{G}^t,\mathbf{s}^t)p(\mathbf{s}^{t+1}|\mathbf{s}^t,\mathcal{G}^t,\mathbf{a}^t). \tag{2.9}$$

由于第 $t+1$ 的状态完全由第 t 步的状态和动作所决定，而不依赖于策略参数 θ 和 φ，因此 $p(\mathbf{s}^{t+1}\,|\,\mathbf{s}^t,\mathcal{G}^t,\mathbf{a}^t)$ 和 $p(\mathbf{s}^1)$ 相对于 θ 和 φ 的梯度为 0。$\nabla_{\theta,\varphi}\mathcal{J}(\theta,\varphi)$ 的完整展开式如下：

$$\begin{aligned}\nabla_{\theta,\varphi}\mathcal{J}(\theta,\varphi) &= \mathbb{E}_{\tau\sim\pi_{\theta,\varphi}}\, R(\tau)\nabla_{\theta,\varphi}\log\pi_{\theta,\varphi}(\tau),\\ &= \mathbb{E}_{\tau\sim\pi_{\theta,\varphi}}\, R(\tau)\nabla_{\theta,\varphi}\sum_t\Big[\log\pi_\theta\left(\mathcal{G}^t|\mathbf{s}^t\right)+\log\pi_\varphi\left(\mathbf{a}^t|\mathcal{G}^t,\mathbf{s}^t\right)\Big],\\ &= \mathbb{E}_{\tau\sim\pi_{\theta,\varphi}}\, R(\tau)\sum_t\Big[\nabla_{\theta,\varphi}\log\pi_\theta\left(\mathcal{G}^t|\mathbf{s}^t\right)+\nabla_{\theta,\varphi}\log\pi_\varphi\left(\mathbf{a}^t|\mathcal{G}^t,\mathbf{s}^t\right)\Big].\end{aligned} \tag{2.10}$$

考虑到 τ 的搜索空间很大，无法全部枚举。为了缓解这一问题，LANE 采用基于蒙特卡洛采样的策略梯度算法 REINFORCE[96] 来估计公式 (2.10)。在实践中，REINFORCE 算法通过从策略分布 $\pi_{\theta,\varphi}$ 中采样 N 个 τ 来近似代替公式 (2.10)中的 $\mathbb{E}_{\tau\sim\pi_{\theta,\varphi}}$。同时，为了降低采样的方差，LANE 采用了带基线奖励的 REINFORCE 算法[97]，其中基准线定义为所有采样得到的 τ 的均值奖励。最终，LANE 用于优化的公式如下：

$$\nabla_{\theta,\varphi}\mathcal{J}(\theta,\varphi)=\sum_{i=1}^{N}(R(\tau_i)-\frac{\sum_{k=1}^{N}R(\tau_k)}{N})\sum_t\Big[\nabla_{\theta,\varphi}\log\pi_\theta\left(\mathcal{G}_i^t|\mathbf{s}_i^t\right)+\nabla_{\theta,\varphi}\log\pi_\varphi\left(\mathbf{a}_i^t|\mathcal{G}_i^t,\mathbf{s}_i^t\right)\Big], \tag{2.11}$$

其中随机变量 τ_i 服从策略分布 $\pi_{\theta,\varphi}$。

差异化更新　组合模块和解算模块的地位是不对称的，因为解算模块要依赖于组合模块的输出。因此，组合模块的更新速度不应与解算模块一样，否则解算模块无法匹配组合模块的输出。受此启发，LANE 引入了一个差异化更新的策略，通过为组合模块和解算模块指定不同的学习率来差异化地更新这两个模块。将 $R(\tau_i)-\frac{\sum_{k=1}^{N}R(\tau_k)}{N}$ 用 r_i 简化表示，组合模块的参数 θ 和解算模块的参数 φ 可以通过下式优化：

$$\begin{aligned}\theta &\leftarrow \theta+\alpha\cdot\sum_{i=1}^{N}r_i\sum_t\nabla_\theta\log\pi_\theta\left(\mathcal{G}_i^t|\mathbf{s}_i^t\right),\\ \varphi &\leftarrow \varphi+\beta\cdot\sum_{i=1}^{N}r_i\sum_t\nabla_\varphi\log\pi_\varphi\left(\mathbf{a}_i^t|\mathcal{G}_i^t,\mathbf{s}_i^t\right),\end{aligned} \tag{2.12}$$

其中解算模块的学习率 β 比组合模块的学习率 α 更高。下文将详细介绍高层策略分布 π_θ 与底层策略分布 π_φ 的具体实现，为简洁起见，以下将 $\mathbf{s}_i^t,\mathcal{G}_i^t$ 和 $\mathbf{a}_i^t$ 相应简化为 $\mathbf{s},\mathcal{G}$ 和 $\mathbf{a}$。

高层策略分布 π_θ　给定 $\mathbf{s}$，组合模块依据参数为 θ 的高层策略分布 $\pi_\theta(\mathcal{G}\,|\,\mathbf{s})$ 采样出 $\mathcal{G}$。如2.2.2节所述，$\mathcal{G}$ 的选取需要轮流应用合并过程和检查过程，自底向上地构建一棵二叉树。假定当前组合模块处于二叉树的第 l 层，记该层时合并过程和检查过程的动作分别

为 $\mathcal{M}_l$ 和 C_l。显然，$\mathcal{G}$ 可以被拆解为一个由 $\mathcal{M}$ 和 C 组成的序列，比如 $(\mathcal{M}_1 C_1 \cdots \mathcal{M}_L C_L)$，其中 L 代表合并过程涉及到的二叉树最高层。记 $\mathcal{M}_l$ 和 C_l 对应的参数分别为 $\theta_{\mathcal{M}}$ 和 θ_C，高层策略分布 $\pi_\theta(\mathcal{G}\,|\,\mathbf{s})$ 可以按照如下公式展开:

$$\pi_\theta\left(\mathcal{G}=(\mathcal{M}_1 C_1 \cdots \mathcal{M}_L C_L)\,|\,\mathbf{s}\right)=\prod_{l=1}^{L}\pi_{\theta_{\mathcal{M}}}\left(\mathcal{M}_l\,|\,\mathbf{s},\mathcal{M}_{<l},C_{<l}\right)\pi_{\theta_C}\left(C_l\,|\,\mathbf{s},\mathcal{M}_{<l+1},C_{<l}\right). \tag{2.13}$$

合并过程的 $\pi_{\theta_{\mathcal{M}}}$ 是由 Tree-LSTM 和全局的可学习的查询向量 $\mathbf{q}$（详情参考2.2.2节）实现的。假定第 l 层有 K 个候选父节点，$\mathcal{M}_l$ 就是一个 K 维的独热向量（One-hot Vector）。在学习时，它是从一个权重为 $(p_1,\cdots,p_K)$ 的 K 维范畴分布（Categorial Distribution）$\pi_{\theta_{\mathcal{M}}}\left(\mathcal{M}_l\,|\,\mathbf{s},\mathcal{M}_{<l},C_{<l}\right)$ 中采样得到的。用 $\mathbf{r}_k^l$ 代表第 k 个候选父节点，它被选择的概率 p_k 是通过在所有合并得分（详情参考2.3节）上归一化计算得到的：

$$p_k=\frac{\exp\left(\left\langle\mathbf{q},\mathbf{r}_k^l\right\rangle\right)}{\sum_{k=1}^{K}\exp\left(\left\langle\mathbf{q},\mathbf{r}_k^l\right\rangle\right)}. \tag{2.14}$$

检查过程的 π_{θ_C} 则是由一个线性层来实现的，其输出的概率遵从期望为 $p_c^l=\sigma(\mathbf{W}_{\mathrm{c}}\mathbf{r}_k^{l+1}+b_{\mathrm{c}})$ 的伯努利分布（Bernoulli Distribution），其中 $\theta_C=\{\mathbf{W}_c,b_c\}$ 是线性层中可学习的参数。值得注意的是，p_c^l 实际上就是2.3节中所提到的 p_c。

底层策略分布 π_φ　当高层代理的动作 $\mathcal{G}$ 确定时，底层代理就被唤起产生动作 $\mathbf{a}$。该动作服从底层策略分布 $\pi_\varphi(\mathbf{a}\,|\,\mathcal{G},\mathbf{s})$，而该策略分布是由带注意力机制的序列到序列网络[27] 实现的：

$$\pi_\varphi(\mathbf{a}=(\mathrm{a}_1\cdots\mathrm{a}_M)\,|\,\mathcal{G},\mathbf{s})=\prod_{m=1}^{M}\pi_\varphi\left(\mathrm{a}_m\,|\,\mathcal{G},\mathbf{s},\mathrm{a}_{<m}\right), \tag{2.15}$$

其中 M 是解码器解码的步数，a_m 代表一个动作词（如 JUMP）或一个程序变量（如 $Y）。在解码的每一步，$\mathrm{a}_m$ 是从一个候选空间为动作词和程序变量的范畴分布中采样得到的。

(2) 奖励函数设计

对于强化学习算法，奖励函数的设计至关重要。LANE 从两个不同的方面考虑了奖励函数的设计，分别是**匹配度**和**复杂度**。值得注意的是，LANE 中的奖励函数是全局性的，也即是说，同一次采样中不同步的动作都使用最终步的奖励进行优化，如图 25中的虚线所示。

基于匹配度的奖励主要依据 LANE 所预测的导航动作序列与真实导航动作序列之间的匹配度得到。由于 LANE 最终输出的是序列，因此匹配度是通过预测与真实序列之

间的交并比（Intersection over Union，IoU）计算得到的。具体地，给定采样输出 $\mathbf{a}^T$ 与真实导航动作 $\mathbf{o}$，基于匹配度的奖励可以通过如下公式计算：

$$R_{\mathrm{s}}(\tau)=\frac{\left|\mathbf{a}^T \cap \mathbf{o}\right|}{|\mathbf{a}^T|+|\mathbf{o}|-|\mathbf{a}^T \cap \mathbf{o}|}, \tag{2.16}$$

其中 $\mathbf{a}^T \cap \mathbf{o}$ 表示序列 $\mathbf{a}^T$ 与序列 $\mathbf{o}$ 之间的最长公共子串，$|\cdot|$ 代表序列的长度。相比于严格匹配，交并比的奖励函数设计缓解了奖励稀疏的问题。

基于匹配度的奖励主要考量最终的结果是否匹配标注，而基于复杂度的奖励更关心得到结果的过程。受启发于奥卡姆剃刀准则**若无必要，勿增实体**，LANE 被期望在学习过程中学到更加简单、更加抽象的解析函数，这样 LANE 就可以泛化到更多的样例上。以 “jump twice” 为例，[“jump twice”→ JUMP JUMP] 和 [“jump” → JUMP, “$x twice”→ $X $X] 这两种方案都可以得到正确的输出。从结果上来看这两种方案无异，但从复杂度上讲第二种方案让 LANE 学习到了适用于其他样例的解析函数 “$x twice”→ $X $X，因此第二种方案是 LANE 更倾向于学习到的。为了将关于复杂度的归纳偏置加入到模型的学习中，LANE 构造了奖励函数 $R_{\mathrm{a}}(\tau)=T^*/T$ 作为模型预测过程复杂度的衡量，其中 T^* 表示 ProLE 中仅包含程序变量的步数。直观地，$R_{\mathrm{a}}(\tau)$ 越小，模型所学习到的方案越难复用，模型预测的过程也就越复杂。结合基于匹配度的奖励和基于复杂度的奖励函数，LANE 中使用的奖励函数 $R(\tau)$ 可用下式表示:

$$R(\tau)=R_{\mathrm{s}}(\tau)+\gamma \cdot R_{\mathrm{a}}(\tau), \tag{2.17}$$

其中 γ 是超参数。在训练中，LANE 的优化目标即最大化奖励函数 $R(\tau)$。

2.5.2 训练策略

由于缺乏相关抽象表达式的标注数据，LANE 采用了分层强化学习算法，通过在候选空间中采样来对组合模块和解算模块进行联合优化。然而，由于候选空间非常大，LANE 模型并不容易收敛。尤其是解算模块，在随机初始化后它有几率采样出无限长的序列，严重影响模型的训练速度和稳定性。为了让训练过程更加稳定，LANE 采用了课程学习（Curriculum Learning）[98] 的训练策略，即从易到难地安排训练样例。在实践中，LANE 根据输入自然语言语句的长度，将训练分成不同的课程阶段。模型首先在简单的课程中训练，模型在该课程上收敛后再增加课程的复杂性。此外，参考前人工作[99]，当训练进入新课程时，LANE 累积旧课程的训练数据以避免灾难性遗忘。

2.6　实验与验证

本节首先介绍用于评估模型性能的评测基准，接着通过详尽的实验验证 LANE 的有效性，最后通过实验分析模型设计每个部分的重要性，并使用两个样例具体展示 LANE 所学习到的抽象表达式。

2.6.1　实验设置

评估任务　如2.1节所述，系统性泛化和生产性泛化是组合泛化的两个重要组成部分，因此，研究者们主要从这两方面来设计任务评估语义解析模型的组合泛化性能。由于对语义解析模型的组合泛化相关研究仍处于起步阶段，前人工作[88, 100] 大都采用人工合成的数据集来评估模型在系统性和生产性上的泛化性能，SCAN[88] 是其中最经典的评测基准之一。如 SCAN 上，研究者们精心设计了若干个任务来评估模型在系统性和生产性上的泛化性能，不同的任务对应着不同的数据集划分。

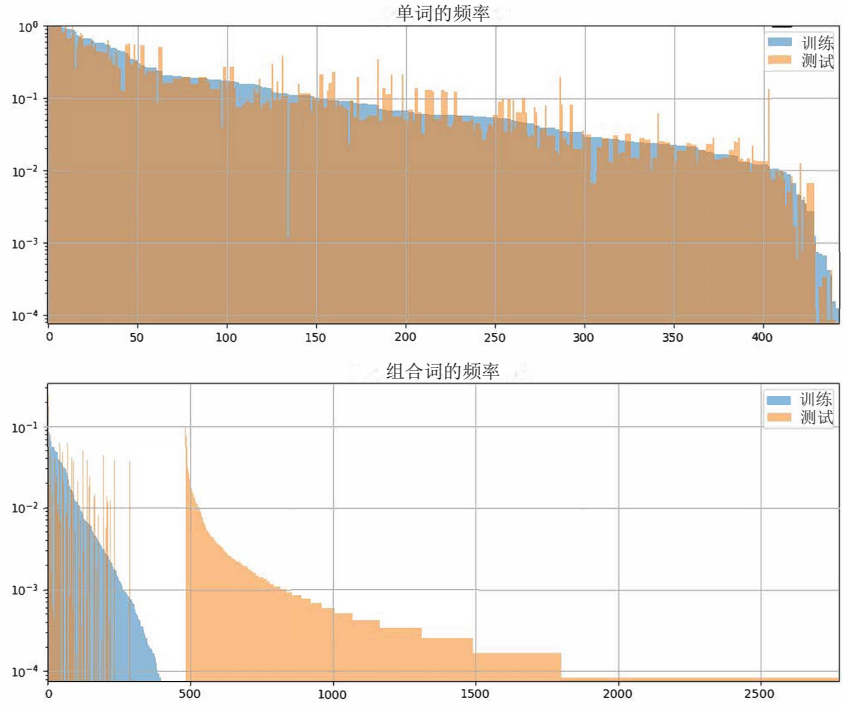

图 26　MCD 任务中训练集和测试集上的组合词分布差异示意图[90]

为评估系统性泛化性能，Lake 等[88] 提出了三个 SCAN 上的任务: (i) ***Add Jump***: 训练集中 “jump” 只单独出现、没有相关的组合性语句，而测试集都是关于 “jump” 的复杂组合。(ii) ***Around Right***: 训练时模型从未见过包含 “around right” 短语的句子，而测试时模型需要理解各种包含 “around right” 短语的组合性语句。(iii) ***Length***: 训练时模型从未见过对应导航动作序列较长的语句（即导航动作序列长度超过 24），但测试时需要模

表 2　实验中用到的数据集信息统计

数据集	SCAN					SCAN-ext	MiniSCAN
任务	Simple	Add Jump	Around Right	Length	MCD (1/2/3)	Extend	Limit
训练数量	16,728	14,670	15,225	16,990	8,365	20,506	14
测试数量	4,182	7,706	4,476	3,920	1,045	4,000	8

型能够理解这些语句。最近，不同于上述三个通过手动划分数据集来构建泛化挑战的任务，来自谷歌的科学家们通过一种自动划分数据集的方法，即基于分布的系统性泛化评估[90]，构造了 ***MCD*** 系列任务。如图 26所示，该任务的训练集和测试集之间有的组合词分布差异很大，对模型的泛化能力提出了更高要求。

为评估生产性泛化性能，本文在 SCAN 的基础上提出了 ***Extend*** 任务。不同于 SCAN 在一句话中最多使用单个 “and” 来连接不同语义，***Extend*** 任务通过增加 “and” 的数量来扩展自然语句的长度范围，比如句子 “jump and walk twice and turn left”。具体而言，训练集中的自然语言至多包含 2 个 “and” 连接词，但测试集中至多允许有 9 个 “and” 同时出现在自然语言中。

表 2展示了用于评估 LANE 的 7 个任务，其中 SCAN 数据集[88] 上的 *Add Jump*，*Around Right*，*Length* 与 MCD (1/2/3) 任务用于评估 LANE 的系统性泛化能力，SCAN-ext 数据集上的 *Extend* 任务用于评估 LANE 的生产性泛化能力。值得注意的是，MCD 算法在为原始数据集划分训练集和测试集时带有随机性，因此不同的随机种子 1，2 和 3 分别对应了不同的任务即 MCD 1，MCD 2 和 MCD 3。除此之外，实验中还引入了 *Simple* 任务用于测试 LANE 在不需要任何组合泛化能力时的性能，以及 MiniSCAN 数据集[101] 中的 *Limit* 任务来测试模型是否能在给定极小样本（例如 14）的情况下学习到组合泛化能力。

基线模型　在 SCAN 的组合泛化挑战上最先进的模型被选择作为 LANE 的基线模型。根据是否使用了人工规则，它们可以分成两组。第一组基线模型**没有使用任何人工规则**，其中包括了经典的带注意力机制的序列到序列模型 Seq2Seq[88, 102]，卷积序列到序列模型（Convolutional Sequence to Sequence，ConvSeq2Seq）[89]，Transformer 模型[103]，Universal Transformer 模型[104]，在注意力机制中融入句法知识的 SynAtt 模型[50] 和通过原子替换实现组合泛化的 CGPS 模型[51]。第二组基线模型**使用了人工规则**，其中包括了以组合泛化为中心的数据增强方法 GECA[56]，结合元学习的序列到序列模型（Meta Seq2Seq）[58]，引入置换等变的序列到序列模型 Equivariant Seq2Seq[57] 和利用程序合成来实现组合泛化的方法（Program Synthesis）[55]。值得注意的是，本节所说的**使用人工规则**是指

表 3　不同模型在 SCAN 系统性泛化任务上的准确率

人工规则	模型	Simple	Add Jump	Around Right	Length
未使用	Seq2Seq[88, 102]	99.7	1.2	2.5 ± 2.7	13.8
	ConvSeq2Seq[89]	100.0	69.2 ± 9.2	56.7 ± 10.2	0.0
	Syntactic Attention[50]	100.0	91.0 ± 27.4	28.9 ± 34.8	15.2 ± 0.7
	CGPS[51]	99.9	98.8 ± 1.4	83.2 ± 13.2	20.3 ± 1.1
	LANE（本方法）	**100.0**	**100.0**	**100.0**	**100.0**
使用	GECA[56]	-	87.0	82.0	-
	Meta Seq2Seq[58]	-	99.9	99.9	16.6
	Equivariant Seq2Seq[57]	100.0	99.1 ± 0.0	92.0 ± 0.2	15.9 ± 3.2
	Program Synthesis[55]	**100.0**	**100.0**	**100.0**	**100.0**

除了原始训练数据外，额外利用了其他数据集特定的人工规则。例如，GECA 方法使用启发式的方法重新组合了训练集而构建了额外的数据，而 Meta Seq2Seq 则采用随机分配原始自然语言对应导航动作的规则来合成额外的训练数据。类似，Program Synthesis 方法引入了人为定义的元语法（Meta Grammar），而这非常依赖数据集。

实现细节　LANE 模型使用 PyTorch 实现[105]，并在所有实验中共享同一套超参数。组合模块中的 Tree-LSTM 包含了一组维度均为 128 的词嵌入、隐藏状态和变量表征。LANE 模型的所有参数都是随机初始化的，并通过 AdaDelta[106] 优化器进行更新，其中组合模块的学习率 α 被设置为 0.1，而解算模块的学习率 β 被设置为 1.0。同时，参考 Havrylov 等[107] 的工作，LANE 在分层强化学习算法中引入了一个 L_2 正则项，以防止模型在训练早期过拟合。该正则项的权重在训练刚开始时被设计为 0.1，并随着课程难度的增加而以 0.5 的倍速指数级下降。实验中 LANE 模型在单个 Tesla-P100（16GB）上进行训练，单次运行的训练时间约为 20 到 25 小时。

2.6.2　实验结果

实验 1. SCAN 上的系统性泛化　如表 3所示，LANE 在所有任务上都达到了 100% 的测试准确率。与没有使用人工规则的最先进的基线模型相比，LANE 实现了显著的性能提升。同时，LANE 的实验结果也非常鲁棒，因为表 3中 LANE 的实验结果是由 5 轮随机实验结果取平均得到，而这 5 轮实验中 LANE 均取得了 100% 的测试准确率。即使与使用人工规则的基线模型（如 Equivariant Seq2Seq）相比，LANE 的性能也极具竞争力，这表明在某种程度上 LANE 能够学习到一定的人工规则。值得说明的是，尽管 Program

表 4　不同模型在 SCAN 基于分布的系统性泛化任务上的准确率

模型	MCD1	MCD2	MCD3
Seq2Seq[90]	6.5 ± 3.0	4.2 ± 1.4	1.4 ± 0.2
Transformer[90]	0.4 ± 0.2	1.6 ± 0.3	0.8 ± 0.4
Universal Transformer[90]	0.5 ± 0.1	1.5 ± 0.2	1.1 ± 0.4
CGPS[51]	1.2 ± 1.0	1.7 ± 2.0	0.6 ± 0.3
LANE（本方法）	**100.0**	**100.0**	**100.0**

Synthesis 方法也能达到很好的性能，但它在很大程度上依赖于预定义的元语法，而元语法或多或少编码了任务相关的知识。LANE 是首个无需任何人工规则就解决 SCAN 数据集上所有系统性泛化任务的神经网络。

实验 2. SCAN 上基于分布的系统性泛化　在更具挑战的基于分布的系统性泛化任务中，LANE 也达到了 100% 的测试准确率（表 4）。从表 3和表 4的结果不难发现，LANE 在任何组合泛化任务上都稳定地保持很好的准确率性能，然而基线模型如 CGPS 会在某些任务上出现大幅的性能下降。同时，LANE 依然是首个解决 SCAN 基准上基于分布的系统性泛化任务的神经网络。

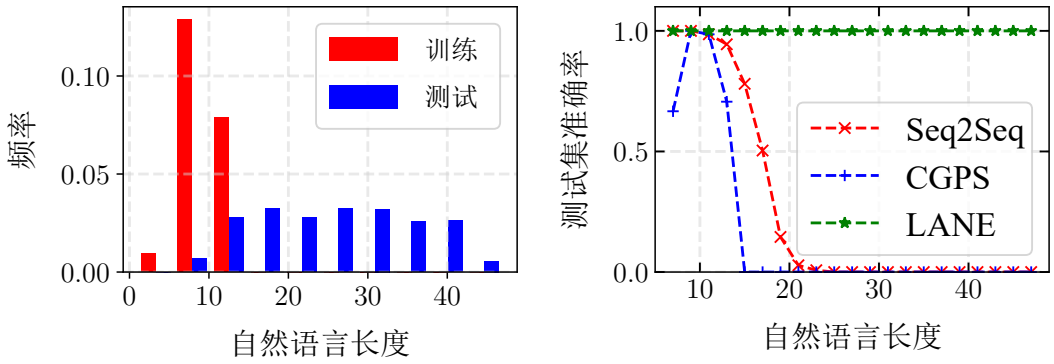

图 27　Extend 任务训练集和测试集上的自然语言长度分布（左）和不同长度上各种模型的测试准确率（右）

实验 3. SCAN 上的生产性泛化　如图 27所示，为了测试生产性泛化性能，Extend 任务控制其训练集和测试集上自然语言长度的分布差异较大。从右侧的测试结果可以发现，基线模型的测试准确率主要由训练集中输入长度所占的频率所决定。相反，随着输入长度的增加，LANE 的测试结果保持着很好的趋势，表明它具备相当的生产性泛化能力。此外，这种趋势亦表明了 LANE 在处理较长的自然语言方面具有较大潜力。

表 5　不同模型在 MiniSCAN 的 Limit 任务上的准确率

模型	Limit
Seq2Seq[101]	2.5
CGPS[51]	76.0
人类水平[101]	84.3
Meta Seq2Seq[58]	100.0
LANE（本方法）	**100.0**

实验 4. MiniSCAN 上小样本条件下的组合泛化　表 5显示了在小样本条件下各种模型的表现，其中人类水平的性能是由 Lake 等[101] 邀请志愿者在有限时间内完成测试估算的性能。从表格中可以发现，即使是在极端少的训练样本条件下，LANE 模型依然非常有效。即使没有任何额外数据，如 Meta Seq2Seq 中采用的基于排列组合的数据增强方法，本方法表现也非常好，在 Limit 任务上的测试准确率为 100%。与人类的表现 84.3% [101] 相比，LANE 的性能也毫不逊色。这在一定程度上表明 LANE 接近了人类可以从小样本中学习组合泛化的能力。但是，值得注意的是，这并不意味着 LANE 在组合泛化的挑战中超过了人类，因为 Limit 是一个相对简单的任务。

2.6.3　实验分析

(1) 消融分析

表 6　SCAN 上所有任务下消融实验的结果

消融组件	Simple	Add Jump	Length	Around Right	MCD1	MCD2	MCD3
层次化抽象	98.5 ± 0.6	0.0	11.1 ± 13.1	0.0	5.3 ± 2.4	0.7 ± 0.3	2.6 ± 0.9
课程学习	0.0	0.0	0.0	0.0	0.0	0.0	0.0
复杂度奖励	100.0	100.0	100.0	0.0	100.0	100.0	78.8 ± 4.2

表 6列举了详尽的消融实验结果，以验证 LANE 模型中每个组件的有效性。首先，层次化抽象过程被消融，即只允许 LANE 抽象并解码一次，意味着 LANE 退化为一个树到序列（Tree to Sequence）的变体模型。该变体首先通过 Tree-LSTM 为每个自然语言建立一棵树并相应地进行编码，但仅在根结点处会触发解算模块并直接生成最终的导航动作序列。正如结果所显示的，层次化抽象的省略会导致 LANE 巨大的性能下降，表明层次化抽象对于 LANE 组合泛化性能的重要性。

接着，课程学习策略被消融，意味着 LANE 从一开始就在所有的训练样本上进行学习。不出意外地，当没有课程学习时，即使训练时间超过 72 小时 LANE 也没有任何收敛的迹象，因此所有任务上 LANE 的性能都降为 0.0。经过分析可以发现，LANE 之所以显示出这种不收敛性，是因为其动作空间极其大。一方面，解算模块有可能输出无限长的序列。另一方面，组合模块的二叉树候选数量也比较多。如此巨大的空间意味着 LANE 能收到的奖励是很稀疏的，特别是对于非常难的自然语言而言。因此，在没有课程学习的情况下，一个随机初始化的 LANE 模型在大多数样例上收到的奖励始终为零。相比之下，通过将样例从易到难排列，课程学习可以极大地缓解了奖励稀少的问题。一方面，容易的样例更有可能提供非零奖励，以帮助 LANE 模型收敛；另一方面，在容易的样例上训练较好的 LANE 模型更有可能在困难的样例上获得非零奖励。

最后，基于复杂度的奖励被消融，意味着 LANE 的奖励函数中仅有基于匹配度的奖励。从表格中不难发现，当基于复杂度的奖励去掉后，LANE 模型在一些任务上做得不好，比如 Around Right 任务。这可能是因为当不鼓励 LANE 尽量采用简单的方案时，LANE 无法自发地从数据中学习到更可泛化的策略。

(2) 变体分析

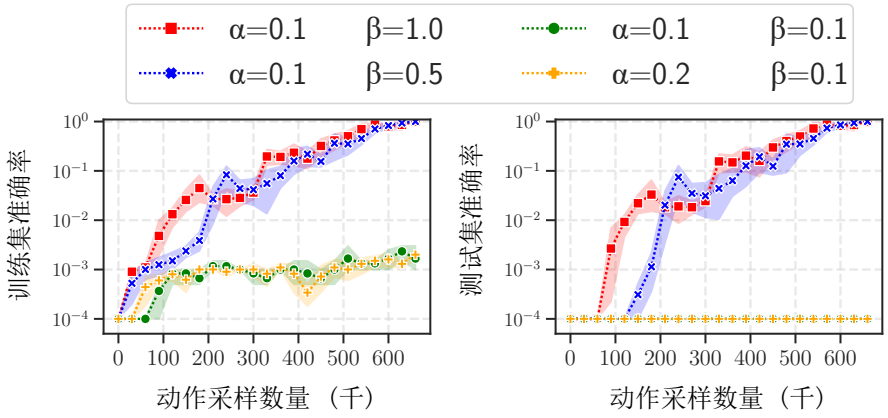

图 28　在不同的学习率组合下，Extend 任务中训练集（左）和测试集（右）上 LANE 模型的准确率

图 28中展示了差异化更新（2.5.1节）中不同学习率的组合下模型的实验结果。从图中可以看出，当组合模块和解算模块的学习率保持一致时，LANE 模型无法收敛，表明了分层强化学习中差异化更新策略的重要性。同时，差异化更新策略并不是只在特定的超参数搭配下才能成功，只要组合模块的学习率比解算模块小即可。

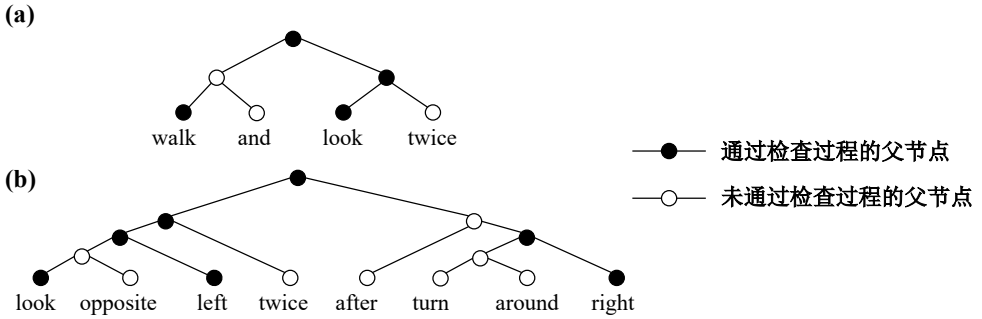

图 29　在两个真实样例上组合模块学习到的树状归纳

(3) 样例分析

为了更直观地展示组合模块所学习到的内容，图 29展示了在两个真实样例上组合模块学习到的树状归纳。可以看到，在不同的上下文中 “twice” 的合并优先级并不相同，这种情况下组合模块能区分对待 “twice” 实属不易。

2.7　本章小结

在本章中，为提升语义解析模型面向程序的组合泛化能力，受启发于人类的层次化抽象思维，本章提出通过学习抽象表达式来改进已有的序列到序列模型。具体地，本章在序列到序列模型的基础上提出了一个记忆单元增强的神经网络架构 LANE，它通过组合模块与解算模块的合作迭代得到最终的程序输出。由于该架构中模块之间传递的信息是离散的，无法直接使用反向传播优化。为了解决模型的训练问题，本章提出应用分层强化学习算法和课程学习策略，成功且稳定地训练了该架构。在组合泛化著名的评测基准上进行的实验表明，该架构以 100% 的测试准确率解决了前人所提出的组合泛化挑战。该架构也是首个不需要任何人工规则就可以全方位应对组合泛化挑战的基于神经网络的方法。

局限性与未来工作　虽然本方法在 SCAN 评测基准上取得了很好的效果，它也存在一些局限性。一方面，本方法的基本假设是自然语言是高度组合化的，即局部片段足够用于表达非常复杂的语义。然而，在自然语言理解需要语义长距离依赖的场景下，该假设是不成立的，这种场景也就无法使用解耦的架构来解析自然语言。另一方面，本方法在实验时忽略了自然语言多歧义的特点，使用上下文无关的表征来构建组合模块。而在现实场景中，自然语言的语义会依赖于其所处的上下文，本方法在这种场景下效果不好。未来工作是将本方法推广以解析有歧义的和有长距离依赖的自然语言。

第三章　面向知识库领域泛化的自然语言语义解析方法

针对领域标注种类少的难点，为提升语义解析模型的泛化能力，本章提出一种无需额外标注训练实体链接模型，并用于强化语义解析模型的方法，显著提升了语义解析模型面向知识库领域泛化能力。

3.1　引言

真实世界中用户的背景是多种多样的，这也就意味着人机之间的互动会在各种领域（Domain）背景下发生，比如金融领域、销售领域和教育领域等等。而在不同的领域中，由于领域风格的不同，用户端所提供的知识库在结构和内容上存在很大差异。例如，金融领域对数字极其敏感，因此其知识库中存在大量的数字内容，而这一特点却不适用于教育领域。近些年，尽管语义解析有了长足的发展，大部分前人工作仍会限定语义解析器模型的使用范围，模型往往仅能处理单个领域场景。即使后来的研究者们开始尝试标注跨多个领域的数据，但因为语义解析标注数据数量弱的问题，模型训练时见过的领域远无法满足真实场景需求。因此，语义解析数据的领域标注种类少成为开发跨领域语义解析模型的难题之一。

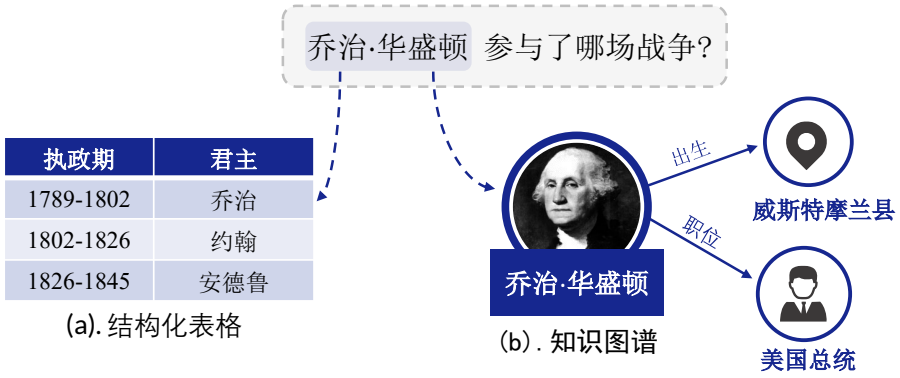

图 30　实体链接示意图，此例中“乔治·华盛顿”分别链接到了表格和知识图谱中

面对该难题，研究者们建议将语义落地机制引入语义解析模型中以增强语义解析模型的领域泛化性。语义落地机制是自然语言处理中的一个经典话题，它通常会将语言符号与现实中的感知或动作相关联[108]。例如，在视觉问答（Visual Question Answering）任务中，给定一个自然语言问句“地上的消防栓是什么颜色”和一张图片，一个好的语义落地机制会将短语“地”与图片中的地面、短语“消防栓”与图片中的对应物体分别关联。语义落地机制被研究者们认为可以提升模型对新场景的泛化性，尤其是面对新视频[109]

与新图片[110] 时。在语义解析场景中，语义落地又被称为实体链接，意指将自然语言中的语义片段链接到知识库中的某个实体。以图 30为例，句子中的 “乔治 • 华盛顿” 既可以被关联到结构化表格中的单元格，也可以被链接到知识图谱中的实体。在语义解析中，许多前人工作都使用到了实体链接机制[2, 38, 111-112]，也都证实了实体链接对语义解析模型领域泛化能力的积极影响。然而，尽管已经取得了一些成功，现有的实体链接方法都有一些固有的缺点。早期的相关工作需要人工编写许多规则，如 Reddy 等[113] 依赖于人工收集的高质量词典，而 Guo 等[2] 则需要人工构造用于实体链接的启发式规则。最近的相关工作则尝试使用数据驱动的方式，通过收集实体链接标注数据直接学习实体链接模型[114-115]。不难看出，这两类方法都需要额外的标注成本，很难将实体链接机制大规模地推广以普遍强化语义解析模型的领域泛化能力。

为了缓解上述问题，针对领域标注种类少的难题，本章提出了一种**擦除后唤醒**方法（Erasing-then-Awakening，下文用 ETA 指代）来利用预训练语言模型的实体链接能力，以增强语义解析模型的领域泛化能力。本方法受启发于可解释机器学习中擦除（Erasing）的相关研究[116]，在该研究中研究者们发现图片中每个像素的重要性可以用一个训练好的图像分类器在该像素擦除前后分类置信度的差值来度量。类似地，ETA 首先构造了一个中间任务，该任务基于预训练语言模型训练了一个实体探测分类器。在该分类器的帮助下，ETA 收集到擦除每个单词前后实体探测器置信度的差值，作为单词与实体之间链接程度的软标签。这些软标签可以作为损失的权重，激励 ETA 基于预训练语言模型训练一个实体链接模型。与前人的工作相比，ETA 不需要任何额外的标注成本，因为中间任务所需要的标签数据完全可以从语义解析数据中自动构造。在四个实体链接数据集上的实证研究表明，即使没有任何实体链接的监督信息，ETA 可以学习到与人类标注高度一致的实体链接能力。更重要的是，ETA 所学习到的实体链接模型可以轻松地与已有的语义解析模型结合，显著地提高它们的领域泛化性。在两个非常有挑战的跨域 text-to-SQL 数据集上的结果表明，ETA 可以显著增强语义解析模型的性能，证实了它在提升模型领域泛化性上的有效性。

3.2　总体结构

本节首先介绍了如何应用本章所提出的实体链接模型增强语义解析模型，接着概述了实体链接模型的模型结构。

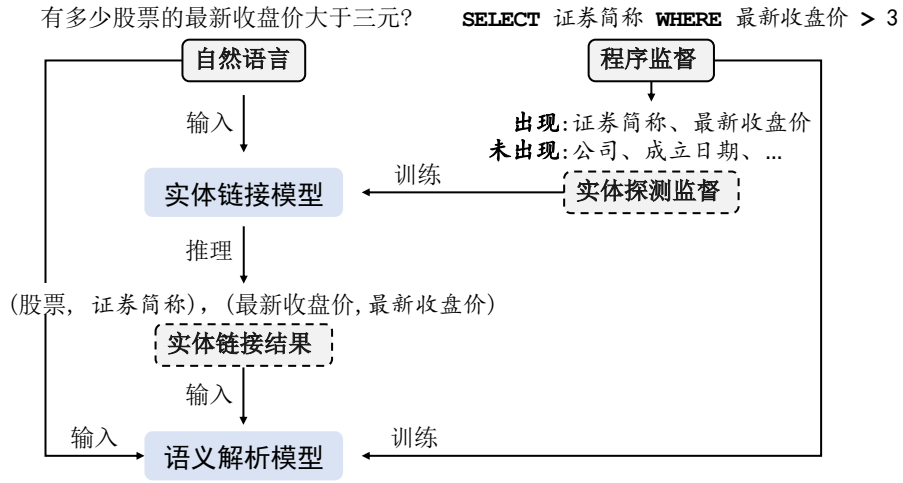

图 31　本章方法示意图：程序监督导出实体探测监督以训练实体链接模型，接着该实体链接模型推理得到的实体链接结果用于增强语义解析模型

3.2.1　方法概述

形式化地讲，给定一个自然语言问句 $\mathbf{x} = \langle x_1, \cdots, x_N \rangle$ 和一个实体集 $C = \{c_1, \cdots, c_K\}$，实体链接模型的目标是找出 $\mathbf{x}$ 中与 C 相关的单词，并将这些单词与它们对应的实体链接在一起。将自然语言问句和实体链接模型推理得到的实体链接结果作为输入，实体链接强化过的语义解析模型的目标是以实体链接结果作为先验信息，生成自然语言问句对应的程序。在本章中，语义解析模型具体是指 text-to-SQL 模型，该模型生成的程序是 SQL 语句。

与前人工作中基于启发式算法的实体链接模型和基于实体链接监督训练的模型不同的是，本章所提出的方法仅需要语义解析的标注数据即可基于预训练模型训练出一个可靠实体链接模型。如图 31所示，本章的方法包含了两个环节：首先，语义解析数据集中标注的程序监督通过确定性的算法转换成了实体探测监督，这些监督信号比程序的信号弱，但可以用在本章所提出的方法中以训练实体链接模型；然后，实体链接模型可以推理出每个自然语言在每个对应知识库上的实体链接结果，这些实体链接结果充当了语义解析模型的先验信息，伴随着语义解析模型的整个训练和推理过程。

3.2.2　模型结构

图 32展示了本章实体链接模型的架构。总体来看，实体链接模型的学习包括了三个步骤：(1) **训练实体探测模块**：通过语义解析任务自动导出的实体探测数据训练一个辅助的探测模块。(2) **擦除问句单词得到软标签**：对自然语言句子进行分词，并挨个擦除，并将每个单词擦除时探测模块的置信度差值作为训练的软标签。(3) **训练链接模块**：将软标签作为监督数据，训练一个编码模块上搭建的链接模块。

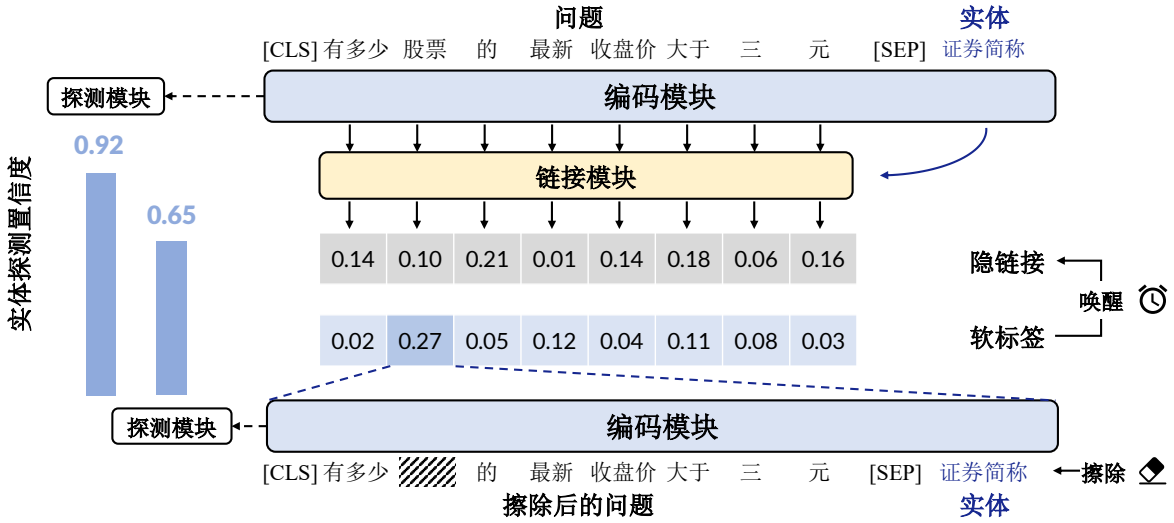

图 32　实体链接模型结构：编码模块、探测模块和链接模块，软标签由擦除问句得到

3.3　实体链接模型学习

本节首先介绍实体链接模型 ETA 的训练过程，然后介绍 ETA 模型的推理过程。

3.3.1　模型训练

在训练时，实体链接模型会学习输出一个形状为 $N \times K$ 的矩阵用于指示自然语言中任一单词与实体集中任一实体之间的链接概率，下文该矩阵被称作**隐链接**（Latent Linking）。

(1) 训练实体探测模块

给定自然语言问句 **x** 和实体集 C，实体探测模块的目标是识别出每个实体 $c_k \in C$ 在自然语言 **x** 中是否被提及。以图 32中的自然语言 “有多少股票的最新收盘价大于三元” 为例，对于实体 “证券简称” 来说，实体探测模块会识别出该实体被该语句提交。对于实体 “城市” 来说，实体探测模块会识别出它并未蕴涵于自然语言的语义中。至于实体探测模块的监督信号，对于实体 c_k 来说记为 $l_k \in \{0, 1\}$，它可以通过语义解析任务自动导出。以语义解析中的经典任务 text-to-SQL 为例，每个样例中都包含一个自然语言语句和一个 SQL 语句。对于任意一个自然语言语句来说，其对应的 SQL 语句中的每个实体（包括表格的表名、列名和单元格值）的语义都蕴涵在自然语言中（$l_k = 1$），而未出现在该 SQL 语句中的、表格中的其他实体的语义则没有体现在自然语言中（$l_k = 0$）。

利用这些监督信号，实体探测模块可以在每个实体表示上构建一个二分类器，该分类器在图 32中示意为**探测模块**。如图 32所示，参考前人的工作[117]，ETA 首先将自然语言问题和所有实体拼接成一个序列送入编码模块。编码模块基于 Transformer[103] 架构，

并通过预训练语言模型 BERT[41] 初始化，它用于为输入序列中的每个单词生成对应的上下文表示。值得注意的是，为简洁起见，图中仅展示了 “证券简称” 实体，实际上编码模块的输入序列通常包含了若干候选实体和一些特殊符号，如用于分隔问题和实体的符号 [SEP]。令 $\langle \mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_N \rangle$ 和 $\langle \mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_K \rangle$ 分别代表问题中每个单词的表征和每个实体的表征，它们可以通过以下公式计算得到：

$$\{\mathbf{q}_n\}_{n=1}^{N}, \{\mathbf{e}_k\}_{k=1}^{K} = \text{PLM}([\text{CLS}], \mathbf{x}, \{[\text{SEP}], c_k\}_{k=1}^{K}), \tag{3.1}$$

其中 $\mathbf{q}_n$ 指的是问题中第 n 个单词的表征，而 $\mathbf{e}_k$ 指的是第 k 个实体的表征。在具体实现时，本文参考前人的工作[117] 使用 c_k 首个单词对应的上下文表示作为 $\mathbf{e}_k$。最终，每个 $\mathbf{e}_k$ 可以通过一个共享的探测模块来得到实体 c_k 的语义蕴涵在问句 $\mathbf{x}$ 中的概率 p_k：

$$p_k = \texttt{Sigmoid}(\mathbf{W}_l \mathbf{e}_k), \tag{3.2}$$

其中 $\mathbf{W}_l$ 是一个可学习的参数。在图 32和下文中，p_k 又被称作**实体探测置信度**。

(2) 擦除问句单词得到软标签

如3.1节所述，本方法受启发于可解释机器学习中的擦除机制，因此 ETA 借鉴了自然语言处理领域应用擦除机制的思路。具体地，ETA 借鉴了前人工作使用单词擦除实现可解释文档分类的思路[118]。在前人工作中，当擦除文档中某个词之后，如果一个训练好的文档分类模型对文档在某个类别上的预测概率有明显的下降，可以认为该词是文档属于该类别的重要特征。在本章的场景中，当擦除问句中某个词之后，如果一个训练好的实体探测模块在某个实体上的探测置信度有明显的下降，可以认为该词与给定的实体密切相关。基于该思路，ETA 使用擦除每个单词后实体探测模块预测分数的差值作为软标签。

擦除单词的具体过程如图 32所示，ETA 会挨个擦除问题中的单词，然后把每个擦除后的问题送入编码模块以及后续的实体探测模块。具体而言，ETA 会用特殊符号 [UNK] 替换掉问句中第 n 个单词 x_n，此时送入编码模块的输入序列可以形式化地表示成 $\langle [\text{CLS}], x_1, \cdots, x_{n-1}, [\text{UNK}], x_{n+1}, \cdots, c_K \rangle$。记擦除 x_n 后实体探测模块对实体 c_k 的置信度为 $\hat{p}_{n,k}$，$\hat{p}_{n,k}$ 和 p_k 之间的差值即可代表 c_k 与 x_n 之间的相关性。将该差值记为 $\Delta_{n,k}$，它可以通过以下公式算得:

$$\Delta_{n,k} = l_k \cdot \max(0, p_k - \hat{p}_{n,k}). \tag{3.3}$$

对问句中的每个单词和实体集中每个实体重复上述过程，ETA 可以得到矩阵 $\Delta \in \mathbb{R}^{N \times K}$，即图 32中用于唤醒隐链接的**软标签**。

(3) 训练链接模块

3.3.1节中描述了如何通过擦除单词得到软标签 Δ，本节将描述 ETA 如何利用这些软标签学习链接模块。在可解释文档分类的前人工作中[118]，研究者们直接利用类似的软标签作为模型的输出，不再进行额外的训练。不同于他们将 Δ 直接作为模型的输出，ETA 引入一个可学习的链接模块来学习 Δ，也就是使用 Δ 作为链接模块训练的软标签。当编码模块输出问句单词 x_n 和实体 c_k 的上下文表示后，构建于编码模块上的链接模块通过以下公式计算两者的链接相关性 $g_{n,k}$:

$$g_{n,k} = \frac{\mathbf{W}_e \mathbf{e}_k \cdot (\mathbf{W}_q \mathbf{q}_n)^T}{\sqrt{d}}, \tag{3.4}$$

其中 $\mathbf{W}_e$ 和 $\mathbf{W}_q$ 是可学习的参数，d 是向量 $\mathbf{e}_k$ 的维度。值得注意的是，$\sqrt{d}$ 的存在是为了归一化 $g_{n,k}$ 的取值范围，最早在 Vaswani 等[103] 的论文中所提出。接着，通过归一化处理（Normalization），链接相关性可以映射到隐链接。记 $\boldsymbol{\alpha}$ 为隐链接，它的每个元素可以由以下公式计算得到：

$$\boldsymbol{\alpha}_{n,k} = \frac{\exp(g_{n,k})}{\sum_i \exp(g_{i,k})}. \tag{3.5}$$

最终，链接模块被训练以最大化似然函数 $\sum_n \sum_k \Delta_{n,k} \cdot \log \boldsymbol{\alpha}_{n,k}$，其中 Δ 被用作似然函数的权重。通过这种方法，ETA 可以从语义解析的粗粒度程序标注中可以学习到细粒度的实体链接模型。

3.3.2　模型推理

在重复擦除单词和训练链接模块这两个过程若干次后，模型的训练会收敛，此时 ETA 可用于实际推理。不同于训练阶段，在推理阶段 ETA 的目标是通过隐链接导出一组**链接对**（Grounding Pair），而每个链接对 $\langle x_n, c_k \rangle$ 表示 x_n 与 c_k 互相关联。注意到实体 c_k 可能会由多个单词组成，ETA 会保留隐链接每列，即 $\boldsymbol{\alpha}_{\cdot,k}$ 中所有超过 $\tau/|c_k|$ 的概率值，其中 τ 是一个阈值，而 $|c_k|$ 是实体 c_k 中单词的数量。另外，考虑到每个自然语言问句中的单词 x_n 都不该链接超过一个实体，ETA 仅保留隐链接每行，即 $\boldsymbol{\alpha}_{n,\cdot}$ 中最高的链接概率。如果 $\boldsymbol{\alpha}_{n,k}$ 未被清空且 $p_k \geq 0.5$，则 $\langle x_n, c_k \rangle$ 被加入最终的链接对中。

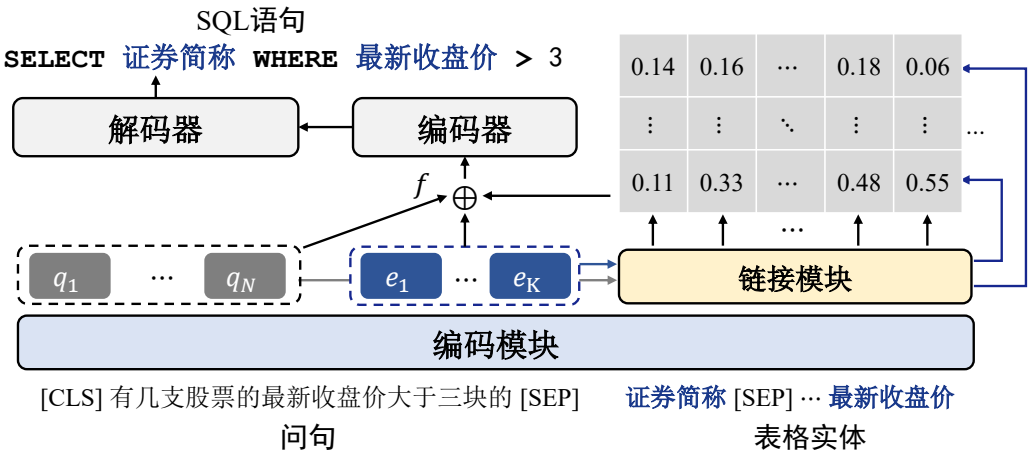

图 33　ETA 与语义解析模型结合框架示意图：编码器接收链接模块的输出作为输入，解码器输出 SQL 语句

3.4　强化语义解析模型

理论上，本章提出的 ETA 是一个即插即用的模型，它可以适用于任何基于知识库的语义解析模型并强化它们的领域泛化性能。为了验证它，本节以 text-to-SQL 作为语义解析的典型样例进行了研究与分析，并提出了一个通用的框架将 ETA 与不同的 text-to-SQL 语义解析器结合。

3.4.1　框架设计

图 33展示了 ETA 与下游 text-to-SQL 语义解析器的结合框架[114]，图中包括了用于获得上下文表示的编码模块，从 ETA 模型中学习到的链接模块和用于生成 SQL 语句的编码器-解码器架构。如1.2.1节所述，目前编码器-解码器架构是语义解析的主流架构，许多最新的语义解析模型都是基于该架构开发。因此，图 33所示的框架适用于许多先进的语义解析模型。例如，图中的编码器和解码器可以被替换为性能更好的模块，只要它们还有编码器和解码器的基本功能，ETA 就可以适应新的模块。

该框架按如下流程运行：

1、首先，给定任意一个自然语言问句和表格中所有实体，模型首先将它们通过3.3.1节的方法拼接起来构成一个输入序列。该输入序列会通过 ETA 模型中的编码模块，从而得到每个问句单词的表示 $\langle \mathbf{q}_1, \cdots, \mathbf{q}_N \rangle$ 和每个实体的表示 $\langle \mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_K \rangle$。

2、接着，将这些表示送入已经学习好的链接模块，从而得到隐链接 $\boldsymbol{\alpha} \in \mathbb{R}^{N \times K}$，即图 33右侧的灰色矩阵。

3、然后，隐链接被用于为自然语言问句中每个单词计算一个实体感知（Entity Aware）的表征。以 $\mathbf{q}_n$ 为例，它对应的实体感知表征 $\tilde{\mathbf{q}}_n$ 通过 $\sum_k \boldsymbol{\alpha}_{n,k} \mathbf{e}_k$ 计算得出。

4、最终，每个实体可感知的表征 $\tilde{\mathbf{q}}_n$ 和原始单词表征 $\mathbf{q}_n$ 拼接在一起送入编码器。随后，解码器通过注意在编码器生成的隐藏状态上，生成 SQL 语句。

通过增加实体感知表征强化编码器，ETA 促使解码器更好地预测 SQL 语句中的实体。

3.4.2　具体实现

在实现时，ETA 会与两个不同的 text-to-SQL 语义解析模型进行结合。第一个是单纯的基于翻译的语义解析模型，使用经典的带注意力的序列到序列架构[26]。具体地，其架构与图 33中所示的架构一样（除链接模块外）。实验使用了 Shi 等[115] 提供的开源代码[1]作为基础语义解析模型，下文中该模型被称为 S2S ，而它结合 ETA 的版本称为 S2S＋ETA 。第二个是利用程序结构的语义解析模型。虽然核心架构依然是带注意力的序列到序列模型，但与 S2S 不同的是，该模型引入了两步解码的 SQL 语句生成策略[28]，在第一步中，它通过一个骨架解码器生成一个粗粒度（Coarse）的 SQL 骨架，相比于具体的 SQL 语句，该骨架中不包含表格实体、聚合函数等信息。在第二步中，它将第一步生成的 SQL 骨架和编码器得到的隐藏状态一起输入到另一个解码器，即语句解码器中，生成目标 SQL 语句。实验使用了 Lei 等[114] 提供的开源代码[2]作为基础语义解析模型，下文中该模型被称为 COARSE ，它结合 ETA 的版本称为 COARSE＋ETA 。

3.5　实验与验证

本节中主要从两个方面评估 ETA ，一方面通过 ETA 在实体链接数据集上的零样本性能评估其是否可以产生与人类专家标注一致的实体链接结果，另一方面通过 ETA 在跨领域语义解析数据集上的性能评估它是否可以提升语义解析模型的领域泛化性能。

3.5.1　实体链接实验设置

数据集　实体链接可以是链接到表格中的实体，也可以是链接到知识图谱中的实体，因此本节选择了两个典型的实体链接子任务，**表格实体链接**和**知识图谱实体链接**，作为评测的任务。对于表格实体链接任务，选择了 SPIDER-L [114] 和 SQUALL [115] 作为评测基准。对于知识图谱实体链接任务，选择了 $\text{WebQSP}_{\text{EL}}$ 和 $\text{GraphQ}_{\text{EL}}$[119] 作为评测基准。值得注意的是，ETA 并未使用这四个数据集上的任何实体链接相关的数据进行训练，这些数据集仅用于评测本方法的性能。

[1]https://github.com/tzshi/squall

[2]https://github.com/WING-NUS/slsql

表 7　实体链接实验使用的数据集统计数据

数据集	训练集		测试集	
	问句数量	实体数量	问句数量	实体数量
SQUALL	9,030	19,185	2,246	4,774
SPIDER-L	7,000	28,848	1,034	4,360
$\text{WebQSP}_{\text{EL}}$	2,974	3,242	1,603	1,806
$\text{GraphQ}_{\text{EL}}$	2,089	2,253	2,075	2,229

评价指标　对于表格实体链接任务，借鉴前人的工作[114]，本节评价了模型在**表格列实体**和**表格表实体**上的精确率（P）、召回率（R）和 F_1 得分。表格列实体和表实体上的 P/R/F_1 分别用 $\text{Col}_P, \text{Col}_R, \text{Col}_F$ 和 $\text{Tab}_P, \text{Tab}_R, \text{Tab}_F$ 来表示。形式化地，令 Ω_{col} 是一个包含 N 个标注链接对的集合 $\{(E_i, Q_i)|1 \le i \le N\}$，其中 E_i 代表第 i 个链接对中的实体，Q_i 代表第 i 个链接对中的自然语言。设 $\overline{\Omega}_{col}$ 是模型的预测结果。类似地，它是一个包含 M 个预测链接对的集合 $\{(\overline{E}_j, \overline{Q}_j)|1 \le j \le M\}$。评价指标 $\text{Col}_P, \text{Col}_R, \text{Col}_F$ 分别通过以下公式得到：

$$\text{Col}_P = \frac{|\Gamma_{col}|}{\left|\overline{\Omega}_{col}\right|}, \tag{3.6}$$

$$\text{Col}_R = \frac{|\Gamma_{col}|}{|\Omega_{col}|}, \tag{3.7}$$

$$\text{Col}_F = \frac{2\text{Col}_P \cdot \text{Col}_R}{\text{Col}_P + \text{Col}_R}, \tag{3.8}$$

其中 $\Gamma_{col} = \Omega_{col} \bigcap \overline{\Omega}_{col}$，即预测链接对和标注链接对的交集。$\text{Tab}_P, \text{Tab}_R, \text{Tab}_F$ 可以通过类似的方法计算得到。对于知识图谱实体链接任务，本节采用前人工作[119] 所提出的实体弱匹配（Weak Matching）的精确率（P）、召回率（R）和 F_1 得分，分别用 $\text{Ent}_P, \text{Ent}_R, \text{Ent}_F$ 表示。该指标的计算方式与 $\text{Col}_P, \text{Col}_R, \text{Col}_F$ 等类似，但因为是弱匹配场景，它放松了判定预测链接对匹配标注链接对的规则。在表格实体链接所采用的指标中，当且仅当预测链接对的实体和自然语言与标注严格匹配时，模型预测的链接对才会被认为预测正确。而在知识图谱实体链接所采用的弱匹配指标中，只要预测的实体正确，且预测的自然语言与标注的自然语言之间存在重叠，模型预测的链接对就被视作是正确的。

基线模型　对于表格实体链接任务，四个先进的方法被选作基线模型：（1）N-gram Matching 方法枚举自然语言问题中所有长度不超过 5 个单词的 N 元短语，并通过模糊

字符串匹配将它们与表格实体联系起来。（2）SIM 方法首先用预训练语言模型编码每个问句单词和每个表格实体，再通过编码得到的上下文表示之间的点积代表每个问句单词和表格实体之间的相关性。（3）CONTRAST 方法借鉴了前人工作[120]，使用对比学习的框架从语义解析的粗粒度监督信息中学习细粒度的实体链接模型。它首先计算每个单词和每个实体之间的相关分数，然后使用最大池化层（Max Pooling）将一个实体在一个自然语言问句上所有的相关分数汇总为一个实体探测分数。最后，一个基于间隔（Margin）的损失被用来鼓励该方法对问句提到的实体给出比未提到的实体更高的实体探测分数。（4）SLSQL_{L} & ALIGN_{L} 都是数据驱动的链接模块，通过完整的实体链接监督数据进行训练。值得注意的是，对于需要阈值的基线模型，它们的阈值都在开发集上的性能相应地进行了调整以进行公平的比较。对于知识图谱实体链接任务，三个先进的方法被选作基线模型：（1）Heuristic 方法通过字符串匹配在知识图谱中寻找候选实体，然后选择出现频次最高的实体作为预测链接实体。（2）VCG 方法混合了不同粒度的实体上下文语境来完成实体链接任务。（3）ELQ 方法使用双塔检索模型进行自然语言和实体之间的匹配，在 $\text{WebQSP}_{\text{EL}}$ 和 $\text{GraphQ}_{\text{EL}}$ 上均取最先进的性能。类似地，VCG 和 ELQ 在训练阶段都需要完整的实体链接监督数据，而 Heuristic 和 ETA 则不需要。

实现细节　ETA 模型架构通过 PyTorch 实现[105]，其中编码模块使用到的预训练语言模型包括了 BERT-base（下文用 ETA＋BERT 指代）和 BERT-large（下文用 ETA＋BERT_{L} 指代），并通过 Transformers 库实现[121]。关于模型优化，模型使用 AdamW 优化器进行优化[122]，每个数据集上的学习率不同。在表格实体链接任务中，模型的学习率是 3×10^{-5}，每次优化使用 24 个样本估计梯度。而在知识图谱实体链接任务中，模型的学习率是 5×10^{-5}，每次优化使用 16 个样本估计梯度。

3.5.2　实体链接实验结果

表 8展示了表格实体链接任务上不同模型的实验结果，ETA＋BERT 大幅度超过了所有未使用实体链接监督数据的基线模型。例如，在 SPIDER-L 上，ETA＋BERT 相比于最好的基线 CONTRAST＋BERT 实现了高达 7.2% Col_F 和 2.8% Tab_F 的提升。

表 9上知识图谱实体链接的实验结果显示出同样的结论。例如，ETA＋BERT 在 $\text{WebQSP}_{\text{EL}}$ 上可以获得高达 74.5% 的 Ent_F ，甚至能与有实体链接监督的基线模型性能持平。

此外，虽然没有在细粒度的实体链接监督下训练，但 ETA＋BERT 在不同的数据集上表现出与部分带实体链接监督数据训练模型相当的性能。例如，在 SPIDER-L 上，本方法在 Tab_F 上比实体链接监督的基线 SLSQL_{L}＋BERT 还要多出 0.9%。在 SQUALL 上，本方法

表 8　表格实体链接任务上不同模型的实验结果，带有 ♡ 的模型在训练时使用了实体链接监督

模型	Spider-L						Squall		
	Col_P	Col_R	Col_F	Tab_P	Tab_R	Tab_F	Col_P	Col_R	Col_F
N-gram Matching	61.4	69.1	65.1	78.2	69.6	73.6	71.6	50.8	59.4
Sim + BERT	16.6	8.0	10.8	8.5	11.6	9.8	13.9	18.0	15.7
Contrast + BERT	83.7	68.4	75.3	**84.0**	76.9	80.3	47.9	31.2	37.8
EtA + BERT （本方法）	**86.1**	**79.3**	**82.5**	81.1	**85.3**	**83.1**	**77.3**	**62.4**	**69.0**
SLSQL_L + BERT ♡[114]	82.6	82.0	82.3	80.6	84.0	82.2	–	–	–
ALIGN_L + BERT ♡[115]	–	–	–	–	–	–	79.2	72.8	75.8

表 9　知识图谱实体链接任务上不同模型的实验结果，带有 ♡ 的模型在训练时使用了实体链接监督

模型	WebQSP_EL			GraphQ_EL		
	Ent_P	Ent_R	Ent_F	Ent_P	Ent_R	Ent_F
Heuristic	30.2	60.8	40.4	-	-	-
EtA + BERT （本方法）	**76.6**	**72.5**	**74.5**	**43.1**	**42.1**	**42.7**
VCG♡ [119]	82.4	68.3	74.7	54.1	30.6	39.0
ELQ + BERT♡ [123]	90.0	85.0	87.4	60.1	57.2	58.6

比起实体链接监督驱动的基线模型 ALIGN_L + BERT 性能也仅略差。注意到 Spider-L 上最好的无实体链接监督的基线模型 Contrast + BERT，在 Squall 上与实体链接监督模型相差甚远。相比起来，但本方法在两个数据集上性能都非常优异。此外，在 WebQSP_EL 和 GraphQ_EL 上，尽管本方法不如最先进的模型 ELQ，但它也取得了与实体链接监督基线模型 VCG 相当的性能。以上结果都证明了本方法所学习到的实体链接模型是可靠的，并与人类专家的标注表现出了高度一致性。

3.5.3　实体链接实验分析

本节通过全面的分析来针对性地回答四个问题：（1）预训练语言模型是 EtA 成功的必要条件吗？（2）是否可以用擦除阶段得到的软标签归一化后的结果直接作为实体链接结果？（3）更大的预训练语言模型是否可以更进一步增强 EtA 的性能？（4）EtA 已有的错误主要有哪些？

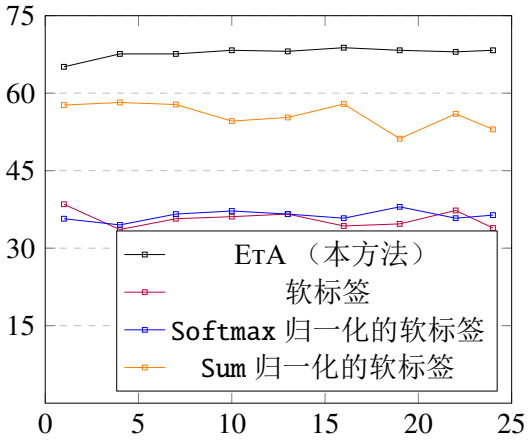

图 34　SQUALL 测试集上不同方法随着训练周期数增加的 Col_F

(1) 预训练语言模型是必要条件

为了探究预训练语言模型对 EtA 的影响，EtA 中原本由预训练模型初始化的编码模块被设置为随机初始化。实验发现，该变体在 SQUALL 上只能得到 40% Col_F ，甚至不如字符匹配算法 N-gram Matching 。考虑到该变体与 EtA＋BERT 其他部分完全一致，它与 EtA＋BERT 之间的巨大差距证明了预训练语言模型是 EtA 方法成功的必要条件。换言之，EtA 的实体链接能力确实是从预训练语言模型中得到的。

(2) 软标签归一化效果欠佳

如3.3.1节中提到的，理论上软标签 Δ 也可以作为实体链接的结果，如果该结论成立的话 EtA 中的唤醒阶段就没有存在的必要。因此，本节实验了软标签的三种变体：原始的软标签，使用 Sum 归一化的软标签以及使用 Softmax 归一化的软标签，这里的归一化指的是将 Δ 矩阵每列的和都归一化为 1.0。如图 34中所示，即使采用各种归一化方法（如 Softmax），Δ 也不能产生令人满意的实体链接性能。相比之下，本方法始终表现良好。通过对 Δ 仔细的分析可以发现，Δ 的值一般都很小，而且彼此之间的差异也不像预期的那样明显。因此，EtA 的成功可能源于唤醒阶段可以鼓励链接模块捕捉到 Δ 中细小的差异，并强化这种差异。

(3) 更大的预训练模型会更好

为探究预训练模型大小对 EtA 的影响，本节探索了使用 BERT-large（$BERT_L$）初始化的模型变体，并在 SPIDER-L 上进行了实验。结果显示 $BERT_{t}extL$ 带来了 2.5% Col_F 和 0.5% Tab_F 的显著改进，这暗示着更大的预训练语言模型可以提供更好的实体链接性能。

表 10　本方法在 SQUALL 数据集上的四种主要错误类型和相应示例

错误类型（占比）	错误样例
遗漏链接（**43.1%**）	How many **points** did arnaud demare receive? 翻译：阿尔诺·德马雷得到多少分？ 标注: **points**→ “UCI world tour points”　　预测: 空
语义正确（**21.0%**）	Total **population** of millbrook first **nation**? 翻译：米尔布鲁克最大的民族有多少人？ 标注: **population**→ “Population” 预测: **population**→ “Population”; **nation**→ “Community”
部分正确（**15.8%**）	Who was **the first winning captain**? 翻译：谁是首个冠军队长？ 标注: **the first**→ “Year”; **winning captain**→ “Winning Captain” 预测: **first**→ “Year”; **winning captain**→ “Winning Captain”
错误链接（**10.1%**）	Were **matinee** and evening performances held **earlier** than the 8th anniversary? 翻译：日场和夜场演出是否在早于 8 周年纪念日时举行？ 标注: **earlier**→ “Date” 预测: **matinee**→ “Performance”; **earlier**→ “Date”

(4) 错误分析

通过仔细检查本方法在 SQUALL 数据集上的错误，本节总结出四种主要的错误类型：(1) **遗漏链接**——本方法没有将任何单词链接到一个实体上；(2) **语义正确**——本方法生成的结果在语义上是正确的，但没有相应的人类标注数据；(3) **部分正确**——本方法没有找到一个实体所有对应的问句单词；(4) **错误链接**——本方法生成了错误的实体链接关系。如表 10中所示，ETA 的错误中只有一小部分是错误链接，这表明 ETA 的精确率较高但召回率较低。

3.5.4　语义解析实验设置

数据集与评价指标　两个跨领域的 text-to-SQL 语义解析数据集被用于实验，包括 SQUALL [62, 115] 和 Spider [10]。借鉴前人的工作，三种评价指标被采用以评测 ETA，分别是**精确匹配**（Exact Match），**集合匹配**（Exact Set Match）和**执行准确**（Execution Accuracy）。前两个评价指标通过比较预测和标注的 SQL 语句来评价模型性能，其中精确匹配通过检查预测的 SQL 语句是否与标注 SQL 语句完全一致来决定预测是否正确，而集合匹配通

表 11　不同方法在 SQUALL 上的实验结果，带有 ♡ 的模型在训练时使用了实体链接监督

模型	开发集		测试集
	精确匹配	执行准确	执行准确
S2S [115]	37.8 ± 0.6	56.9 ± 0.7	46.6 ± 0.5
S2S + BERT [115]	44.7 ± 2.1	63.8 ± 1.1	51.8 ± 0.4
S2S + ETA + BERT （本方法）	**47.6** ± 2.5	**66.6** ± 1.7	**53.8** ± 0.3
ALIGN ♡[115]	42.2 ± 1.5	61.3 ± 0.8	49.7 ± 0.4
ALIGN + BERT ♡[115]	47.2 ± 1.2	66.5 ± 1.2	54.1 ± 0.2

过检查预测的 SQL 语句子句的正确性来评估其正确性。相比于精确匹配，集合匹配不关注 SQL 语句中关键词预测的前后顺序。执行准确则通过比较预测和标注的 SQL 执行的结果来评价模型性能，当预测的 SQL 语句对应的执行结果与标注的标准答案一致时，认为模型预测正确，否则认为不正确。

基线模型　在 SQUALL 数据集上，基线模型包括 S2S 和 ALIGN ，其中前者是经典的基于注意力的序列到序列模型，后者用一个通过实体链接标注数据训练得到的链接模块增强 S2S 。类似地，在 Spider 数据集上，主要的比较基线模型是 COARSE 和它的实体链接强化版 SLSQL ，其中前者是基于两步解码的语义解析模型，如3.4.2节所述。同时，为了进行全面的比较，本节还将 ETA 与 Spider 上最先进的模型 [3]进行了比较，包括使用图神经网络建模实体之间关系的 GlobalGNN[40]，提出中间表示提升语义解析模型性能的 IRNet[2]，对表格实体之间的关系进行细致建模的 RATSQL[38] 和建模问句与表格之间的精细关系的 BRIDGE[124]。

3.5.5　语义解析实验结果

表 11和表 12分别展示了不同方法在 SQUALL 和 Spider 上的实验结果。其中，SQUALL 上不同方法的实验结果是在 5 个随机种子下实验得到的结果均值与标准差，而 Spider 上不同方法的结果是它们在开发集上取得最好性能模型的结果。正如表中所示，引入 ETA 可以极大地提高两个已有的 text-to-SQL 语义解析模型的性能，证明了 ETA 提升下游语义解析模型领域泛化能力的有效性。以 Spider 为例，本方法 COARSE + ETA + BERT 相比于基线模型 COARSE + BERT ，在集合匹配指标上提高了 7.1%。随着预训练模型变得更大（例如，BERT_{L}），ETA 对语义解析模型的泛化性能的改进变得更加明显，最高可以

[3] https://yale-lily.github.io/spider

表 12　不同方法在 Spider 上的集合匹配结果，带有 ♡ 的模型在训练时使用了实体链接监督

模型	开发集	测试集
GlobalGNN + BERT [40]	52.7	47.4
IRNet + BERT [2]	61.9	54.7
BRIDGE + BERT [124]	65.5	59.2
BRIDGE + BERT_L [124]	70.0	65.0
RATSQL + BERT_L [38]	69.7	**65.6**
COARSE + BERT [114]	57.4	-
COARSE + BERT_L [114]	61.0	-
COARSE + ETA + BERT （本方法）	64.5	59.5
COARSE + ETA + BERT_L （本方法）	**70.8**	65.3
SLSQL + BERT ♡[114]	60.8	55.7
SLSQL + BERT_L ♡[114]	65.1	-

达到 9.8% 的绝对提升。与最先进的方法相比，COARSE + ETA + BERT_L 也获得了极其有竞争力的性能。考虑到 ETA 所结合的语义解析模型结构相对简单，能取得与最先进的复杂模型（如 RATSQL + BERT_L ）相当的性能是非常不易的。

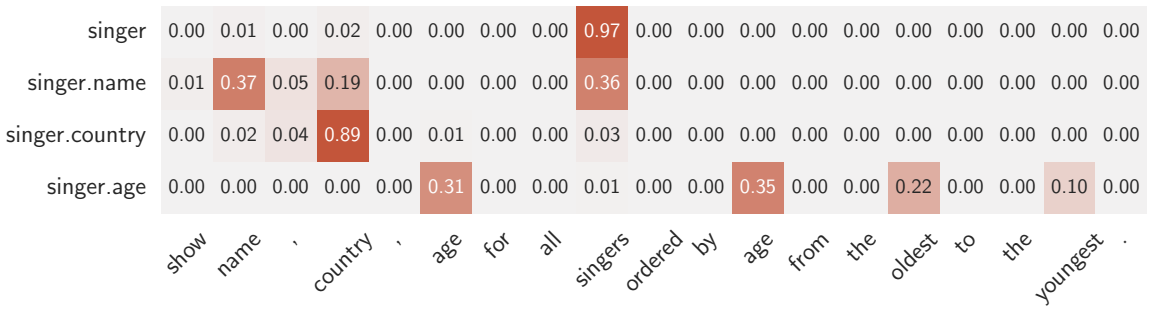

图 35　COARSE + ETA + BERT_L 在真实样例上隐链接的可视化结果，其中横轴是自然语言，纵轴是表格的列名

更有趣的是，在这两个数据集上，本方法与使用实体链接监督的方法相比，可以达到类似甚至更好的性能。例如，与 SLSQL + BERT 在 Spider 上的集合匹配指标相比，COARSE + ETA + BERT 比它高出 3.7%。考虑到 SLSQL 使用了额外的人类标注数据，如此显著的性能提升是非常令人惊讶的，本节将其归因为两个原因：

1、预训练模型已经学会了部分实体链接的能力，而本方法可以成功地将其唤醒。

2、使用实体链接监督的语义解析模型一般都是先使用实体链接数据单独训练一个链接模块，再将其结合到语义解析模型中。但这样的结合方法会引入曝光偏差（Exposure Bias）的问题，因为在训练时语义解析模型使用的是一定正确的实体链接标注，但测试时使用的却是可能出错的实体链接预测。而本方法训练和测试时都使用的是实体链接模型的预测结果，不存在曝光偏差问题。

3.5.6　语义解析实验分析

(1) 可视化分析

图 35展示了 Coarse＋EtA＋$BERT_L$ 在真实问题 “Show name, country, age for all singers ordered by age from the oldest to the youngest”（**翻译：显示所有歌手的姓名、国家、年龄歌手，并按年龄从大到小排序。**）上的隐链接可视化结果，可以看出本方法所构建的隐链接表现出稀疏的特质，即只在部分可能产生链接的地方才有概率分布。

(2) 样例分析

表 13展示了在三个真实样例中，Coarse＋EtA＋$BERT_L$ 的预测结果。表中问题和 SQL 语句中标蓝色的是表格列名，标绿色的是表格表名，问题和 SQL 语句中标记为同一序号的片段互相链接。正如从表格中所观察到的，EtA 在许多语义类型的实体链接场景下表现都很出色，包括形容词类型（如 oldest ↔ age）、实体类型（如 where ↔ hometown）和动词类型（如 registered ↔ student_enrolment）的实体。同时，这些实体链接的结果为用户理解 SQL 语句的生成过程提供了一些可解释性，这可以帮助他们更好地理解语义解析模型的预测过程。

3.6　本章小结

在本章中，为提升语义解析模型面向知识库领域的泛化能力，本章提出了一种通过擦除单词从预训练的语言模型中唤醒实体链接能力的方法。只需要语义解析的监督数据，该方法可以从预训练的语言模型中生成用于实体链接模型训练的软标签。在四个实体链接数据集上的实验结果表明，以软标签作为监督所训练出的实体链接模型效果甚至可以匹配全监督训练出的模型。更重要地，在两个经典的跨域语义解析数据集上的实验结果表明，该实体链接模型可以灵活地应用到现有的 text-to-SQL 语义解析器中，并显著提高它们面向知识库的领域泛化能力。

表 13　COARSE + ETA + BERT$_L$ 在 SPIDER-L 开发集的三个实例上所预测的实体链接与 SQL 语句

序号	实例
问题 1.	Show name$_1$, country$_2$, age$_3$ for all singers$_4$ ordered by age$_3$ from the oldest$_3$ to the youngest. 翻译：按年龄从大到小显示所有歌手的名字、国家和年龄。
SQL 语句 1.	SELECT name$_1$, country$_2$, age$_3$ FROM singer$_4$ ORDER BY age$_3$ DESC
问题 2.	Where$_1$ is the youngest$_2$ teacher$_3$ from? 翻译：最年轻的老师来自哪里？
SQL 语句 2.	SELECT hometown$_1$ FROM teacher$_3$ ORDER BY age$_2$ ASC LIMIT 1
问题 3.	For each semester$_1$, what is the name$_2$ and id$_3$ of the one with the most students registered$_4$? 翻译：哪个学期注册学生最多，列出它的名字和号码。
SQL 语句 3.	SELECT semester_name$_2$, semester_id$_3$ FROM semesters$_1$ JOIN student_enrolment$_4$ ON semesters.semester_id = student_enrolment.semester_id GROUP BY semester_id$_3$ ORDER BY COUNT(*) DESC LIMIT 1

局限性与未来工作　虽然本方法在提升语义解析的领域泛化能力上效果显著，但它也存在一些局限性。一方面，本方法需要两阶段的训练，训练流程比起前人工作较为繁琐。另一方面，本方法仍然需要一些语义解析任务的监督数据才能训练一个实体链接模块，这限制了其在更多任务上的应用。未来工作是将本方法改进成联合训练实体链接模块与语义解析模块，并提出无监督实体链接的方法。

第四章　弱监督下答案驱动的自然语言语义解析方法

针对答案标注监督弱的难点，为提升语义解析模型在弱监督下的性能，本章提出使用生成式模型可微地解决弱监督语义解析的方法，并提出一种执行引导的预训练方法增强其性能，方法效果与强监督下最好的基线模型性能持平。

4.1　引言

语义解析的一大挑战在于语义解析中程序标注的获取，因为需要收集自然语言与对应程序的平行语料。虽然自然语言相对容易获取，但程序的获取需要精通目标程序的专家，收集较为困难。这种困难主要体现在两个方面，数据标注的耗时与数据标注的昂贵。例如，Yu 等[10] 在构造包含约 10,000 个训练样本的 text-to-SQL 数据集 Spider 时邀请了 11 名计算机专业的耶鲁大学本科生，耗费将近 1,000 个小时才收集好数据，数据标注非常耗时。而在 Yin 等[14] 的研究中，text-to-Python 数据集 CoNaLa 的构造遵循了先自动爬取，后人工修订的标注流程。然而，仅是修订形如 shutil.copy('file.txt', 'file2.txt') 这样相对简单的行内程序都需要花费将近 1 美元，可以预见标注复杂程序的成本要更高，数据标注非常昂贵。同时，虽然对于一些常见的程序语言如 SQL 或 Python，找到精通程序的专家相对容易。更多的情况下，许多编程语言较难找到足够多的专家，如 GeoQuery 数据集所使用的较为小众的编程语言 λ-calculs 表达式[30]。

年份	城市	国家	参赛国家数量
1896	雅典	希腊	14
1900	巴黎	法国	24
1904	路易斯	美国	12
…	…	…	…
2004	雅典	希腊	201
2008	北京	中国	204
2012	伦敦	英国	204

问题：希腊在哪一年举行了其最后一届夏季奥运会?

答案：2004

问题：哪个城市举办的奥运会的参赛国家数量首次超过20?

答案：巴黎

问题：哪几年参赛的国家数量最多?

答案：2008, 2012

图 36　弱监督语义解析任务仅提供自然语言问题对应的答案作为监督

考虑到为自然语言标注程序的高昂成本阻碍了语义解析的大规模发展，许多研究者[11, 62, 125] 都在考虑使用成本更低的数据作为**间接监督**（Indirect Supervision）来训练一个语义解析器。因为语义解析最终仍需要通过执行程序满足用户的需求，而最常见的需求是回答用户的问题，因此大量前人工作在探索**仅标注自然语言问句答案作为语义解析**

监督的场景[126-130]，该场景也被称为弱监督语义解析。图 36展现了表格上弱监督语义解析任务的三个示例，模型在训练时只能见到表格、问题与答案，而没有 SQL 语句。从示例中不难发现，与程序标注需要专家细致编码不同，普通人稍加思考就可以推理出自然语言问题对应的答案，因此弱监督语义解析数据标注的金钱成本和时间成本显著更低。根据 Choi 等[131] 的统计数据，从头标注一对问题和答案的成本平均仅需 0.3 美元。不过，虽然弱监督语义解析的数据标注成本降低了，但是模型学习的难度却变高了。考虑到答案标注所蕴含的信息量远小于程序标注，答案标注监督弱就成为弱监督语义解析下的主要难题。面对该难题，前人工作一般采用强化学习的方法，即从语义解析模型中采样程序，并将该程序的执行结果与标准答案的重合率作为语义解析模型奖励优化模型。但这类方法往往面临着虚假程序密集[67]，动作空间奖励稀疏[65] 和模型训练冷启动困难[64] 等问题。最近，也有研究工作采用可微分方法处理弱监督语义解析问题[44]，然而他们的方法为了能够反向传播而牺牲了模型的表达能力，无法应用到复杂的场景。

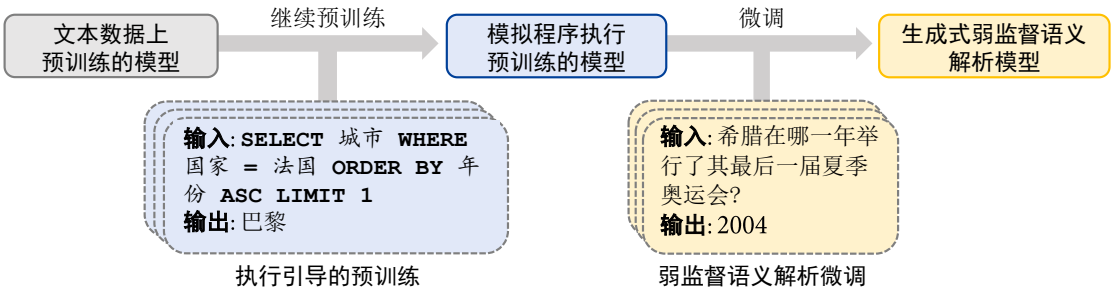

图 37　预训练与微调阶段模型输入输出示意图：预训练阶段模型接收 SQL 程序和表格作为输入，而微调阶段模型接收自然语言和表格作为输入

为缓解以上问题，针对“答案标注监督弱”的难题，本章首先提出一个新的思路来应对弱监督语义解析场景，即将弱监督语义解析视为序列生成任务，并利用生成式语言模型直接建模自然语言问句和对应的知识库，并生成自然语言问句的答案。生成式语言模型既拥有较强的表达能力，又可以通过反向传播稳定优化。然而，由于生成式语言模型仍然遵循序列到序列架构，仍然有数据饥饿的问题，本方法在训练样本数较少时效果较差。因此，为了让生成式模型在数据量较小时仍能有效，本章提出了一个执行引导的预训练方法 TAPEX 来增强模型。不同于传统的预训练方法往往需要大规模地爬取并清洗数据[44-45]，本方法通过合成高质量的预训练语料库来对模型进行预训练。在预训练时，模型通过学习随机复杂程序（如 SQL 语句）的执行过程来间接学会自然语言背后的执行逻辑。如图 37所示，在继续预训练阶段模型模拟 SQL 执行器，学习生成 SQL 语句的执行结果。在微调阶段，预训练好的模型可以用于弱监督语义解析模型的初始化，并在弱监督语义解析场景微调以完成相关任务。为了验证该方法的有效性，本章在三个著名的弱监督语义解析数据集上做了实验。实验结果清楚地表明，在下游数据较多时生成式

弱监督语义解析方法可以媲美前人工作的性能，而 TAPEX 可以带来持续的改进。在下游数据较少时虽然生成式弱监督语义解析方法效果欠佳，但 TAPEX 可以极大地提升其性能，最高可以带来 19.5% 的绝对提升。最终，本方法在所有的实验基准上都取得最先进的结果，以显著的优势超过所有基线模型。更重要的是，本方法可以在其中两个基准上取得与强监督下最好基线模型可比的性能。接下来，本章将主要针对图 36中基于表格的弱监督语义解析场景介绍方法的具体实现，但理论上该方法也可以很容易迁移到其他弱监督语义解析场景，如基于知识图谱的弱监督语义解析。

4.2 总体结构

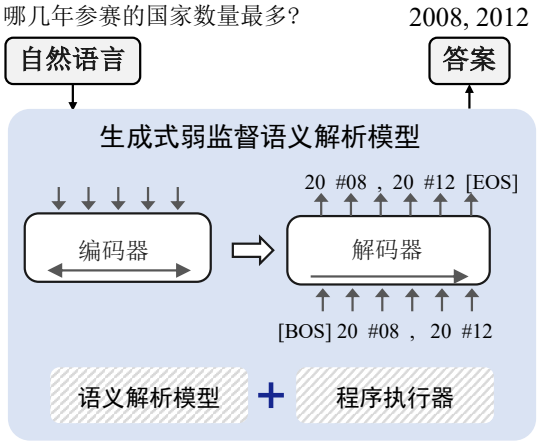

图 38 本章方法示意图：用序列生成的方法处理弱监督语义解析场景，自回归地生成问题对应的答案

如图 38所示，本章所提出的方法将弱监督语义解析视为生成任务，利用一个双向的编码器和一个单向的解码器直接生成自然语言问句的答案。在生成答案时，解码器将答案切分成若干个子词（Subword，如图中的“#12”），并通过自回归（Auto-regressive）解码的方式生成它们，其中 [BOS] 是指示序列开始解码的提示词，而 [EOS] 是指示序列结束解码的提示词。不同于大部分前人工作使用强化学习应对弱监督语义解析，本方法可以采用监督学习的范式进行训练，即模型可以直接通过反向传播优化。同时，由于该架构与生成式语言模型兼容，凭借生成式语言强大的表达能力，该方法可以轻易扩展到任意复杂度的场景，不论答案是一个集合、一个实体还是一个计算结果。直觉上来讲，本方法融合了一个潜在的语义解析器和程序执行器，这两个隐藏的模块搭配完成了弱监督语义解析的任务。

4.3 生成式弱监督语义解析

本节描述了弱监督语义解析任务的形式化定义和生成式模型解决该任务的细节。

4.3.1　形式化定义

在本章所面向的表格为知识库的场景中，弱监督语义解析任务的目标是通过检索表格内容回答用户的问题。因此，该任务的输入通常会包括一个自然语言问题 $\mathbf{x}$ 和一个结构化的行列表 T。每个自然语言问题 $\mathbf{x}$ 由 K 个单词组成，其形式化表示即 $\mathbf{x} = \langle x_1, x_2, \cdots, x_K \rangle$。每个表格 T 都包含 M 行，即 $\{r_i\}_{i=1}^{M}$，其中每一行 r_i 又包含 N 个单元格 $\{s_{(i,j)}\}_{j=1}^{N}$，每个单元格 $s_{(i,j)}$ 又由若干个单词组成，且对应到表格的第 j 列 c_j。该任务的输出往往是一系列单元格值（如图 36中的第二个答案），或者是一个通过聚合函数（如 MAX）在指定单元格区域计算得到的数字（如图 36中的第一个答案）。

4.3.2　方法细节

通过将弱监督语义解析建模为序列生成任务，并利用生成式预训练语言模型通过逐个单词解码生成答案，本章所提方法有如下几个优点:

1、灵活性：由于生成式模型强大的建模能力，该方法可以很容易地适应（几乎）任何种类的输出。

2、便捷性：该方法不需要对预训练好的语言模型进行任何修改，可以直接在预训练好的模型权重基础上进行训练。

3、迁移性：由于弱监督语义解析任务都被建模为序列生成任务，该方法允许不同的数据集使用相同的训练流程，也因此很容易执行多任务学习。

下面将详细介绍模型结构、输入、输出与训练策略。

模型结构　理论上该方法适用于任何生成式语言模型，只要它能够灵活地生成序列，如 GPT3[42] 和 UniLM[132]。在本章的实验中，该方法主要基于目前学术界使用较广泛的预训练生成式语言模型 BART[133] 作为基础模型来实现。BART 模型主体上是一个基于标准 Transformer 架构[103] 的序列到序列模型。与原始架构略有不同的是，BART 将 Transformer 中使用的非线性激活函数从 ReLU 函数替换为 GeLU 函数[134]。在预训练时，不同于 BERT 模型采用的随机单词掩码（Token Masking）[41]，BART 采用的是随机短语掩码（Span Masking），即随机掩盖句子中变长的相邻短语。将单个 [MASK] 特殊符号掩盖每个变长短语后的句子作为输入，BART 试图输出原始句子，并优化句子级重构的损失。通过这样的架构和预训练目标，BART 不仅可以像 BERT 一样通过重建任务学习到深度上下文表征，还可以灵活地生成自然语言。

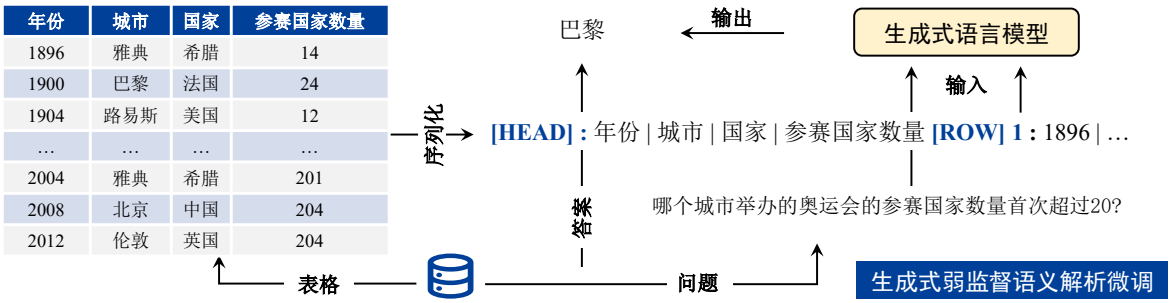

图 39　生成式语言模型解决弱监督语义解析任务示意图：将问题与序列化表格拼接构造的文本输入模型，模型被训练输出问题对应的答案

模型输入　如图 39所示，模型输入包含一个自然语言问句和一个结构化的表格。自然语言问句的编码相对简单，遵从语言模型在预训练阶段对自然语言的处理方法即可。然而，对于模型而言表格的编码并不简单。相比于自然语言的非结构化，表格有着高度结构化的行与列。在实践中，本节借鉴前人工作[135]，直接将表格按行拼接构成一个序列，这个过程在图 39中被标为**序列化**。形式化地，通过一些特殊的单词如 [HEAD] 来表示表格中列、行与单元格的边界，一个序列化的表格可表示为:

$$T^{*} = \langle\, [\text{HEAD}], c_1, |, \cdots, |, c_N, [\text{ROW}], 1, r_1, [\text{ROW}], 2, r_2, \cdots, r_M \,\rangle, \tag{4.1}$$

其中 [HEAD] 和 [ROW] 是特殊的单词，分别表示表头和行的区域，而 [ROW] 后面的数字 1 则表示行的索引。除引入分隔区域的单词外，模型输入还引入了竖线符号 | 用于分隔不同列的表头或单元格，如图 39中的**年份 | 城市 | 国家 | 参赛国家数量**。最后，自然语言问句 **x** 被拼接在序列化的表格 T^{*} 前，组成了模型编码器的输入。下文展示了各个数据集上模型输入的具体样例，具体可见表 14。

模型输出　在 BART 架构中，编码器负责编码模型输入，而解码器负责对模型输出进行建模。如4.3.1节所述，当模型的输出是一个实体或一个数字时，模型将其视作一段文本，通过自回归解码逐个单词生成它。当模型的输出是一个集合时，模型将由逗号分隔拼接在一起的集合整体视作一段文本，如图 36中的 **2008, 2014**。通过将所有答案均视作自由文本，本方法可以支持弱监督语义解析场景下任意操作以及它们的复杂组合。下文也展示了各个数据集上模型输出的具体样例，具体可见表 14。

训练策略　由于本方法在统一的架构上完成各种弱监督语义解析任务，所以它可以很容易地进行多任务学习[43]。因此，本文探索了两种训练模型的策略，一种是单任务训练，另一种是多任务训练。前者是在每个单独的下游数据集上对模型进行微调，而后者是首

先在相关的其他下游数据集上微调模型，然后继续在目标数据集上微调模型。

4.4　执行引导的预训练

4.3节主要介绍了如何在弱监督语义解析任务上微调生成式语言模型，本小节将从预训练任务和预训练语料库两个方面介绍执行引导的预训练方法 TAPEX 。

4.4.1　预训练任务

在自然语言的预训练中，掩码式语言模型通过重建遮盖的单词获得了巨大成功[41]，因此已有的面向弱监督语义解析的预训练工作也常常使用重建任务进行预训练。以本章所关注的表格作为领域知识的弱监督语义解析方向为例，前人工作通常将被随机遮盖的表格和句子联合作为模型输入，通过让模型恢复被遮盖的部分来加强模型对这两部分的联合推理能力[44-45]。虽然重建任务表现良好，但是它的预训练效率往往较低，通常需要一个极其大的预训练语料库完成预训练。

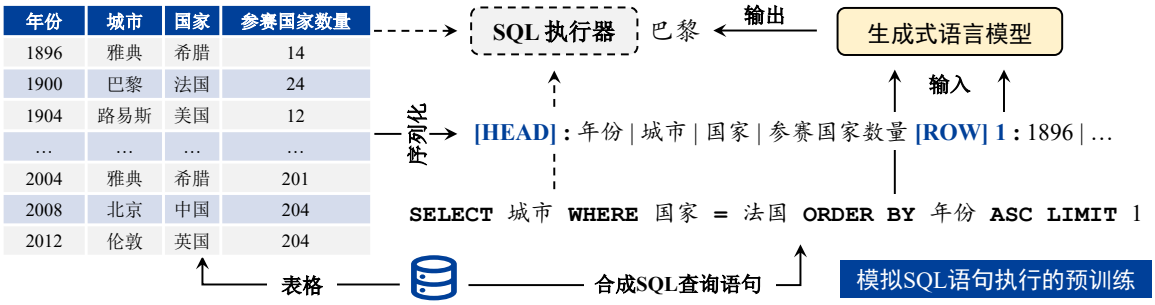

图 40　执行引导的预训练示意图：将随机生成的 SQL 语句与序列化表格拼接构造的文本输入模型，模型被训练输出 SQL 语句对应的执行结果

预训练任务的设计要点在于要靠近下游任务，在本章所面向场景中这种能力是**在表格上进行推理的能力**。回想一下图 36中的第一个问题 “哪几年参赛的国家数量最多?”，即使该问题表述成 “按参赛国家数量排序，取最大，返回年份”，它们所需要的语义操作并没有本质变化，都需要模型在对应表格上排序、取最值、返回年份一列上的投影结果。因此，即使预训练时使用的是随机采样的 SQL 语句，模型也应当能从它的执行过程中学习到自然语言推理的背后逻辑。考虑到这一点，TAPEX 将模仿 SQL 语句的执行过程作为它**唯一**的预训练任务。正如在图 40中所展示的，TAPEX 的预训练与4.3节中生成式弱监督语义解析的过程类似。最大的不同之处在于，预训练时模型的输入从自然语言问句变成了 SQL 语句。形式化地，给定一个可执行的 SQL 语句和一个结构化表格 T，TAPEX 首先将 SQL 查询语句和序列化的表 T^* 拼接起来输入模型编码器。然后，它通过一个成熟的 SQL 执行器（如 MySQL）获得查询的执行结果，作为模型解码器的输出。

可以预想到，如果一个语言模型能够被训练“执行”SQL 语句并产生正确的执行结果，那么它在理解自然语言的语义时也会更加轻松。

4.4.2 预训练语料

如上所述，TAPEX 使用的预训练语料关键因素包括结构化表格和 SQL 语句，因此本小节分别介绍这两部分数据的来源。

(1) 结构化表格选择

前人工作一般使用可公开获取的表格作为预训练的表格来源[44-45]，因此 TAPEX 也使用公开的结构化表格作为预训练语料库的表格来源。然而，与前人工作需要数百万的结构化表格不同，受益于下文中 SQL 查询语句的多样性，TAPEX 只需要几千个表格就可以避免模型在预训练阶段过拟合到特定表格上。

因此，TAPEX 没有采取爬取带噪表格后使用规则清洗的策略，而是直接从已有的公共数据集中收集高质量的表格。具体来说，TAPEX 从公开数据集 WIKITABLEQUESTIONS [62] 的训练集中随机选择了 1,500 张表格作为预训练语料库的表源。

(2) SQL 语句合成

由于 SQL 语言是遵循特定 SQL 语法的，因此它可以系统性地合成。在前人的工作中，合成 SQL 语句的方法主要可以分为两类: （1）利用概率化的上下文无关文法采样生成 SQL 语句[136]；（2）实例化 SQL 模板生成 SQL 语句[137]。TAPEX 采用了后者，即通过实例化 SQUALL 数据集[115] 上自动抽取的 162 个 SQL 模板来合成 SQL 查询语句。SQL 模板是通过将 SQL 语句中表格相关的信息抽象成其对应类别后得到的。例如，SQL 模板 SELECT num_1 WHERE $\text{text}_1 = \text{val}_1$ 中的 num_1 和 text_1 分别对应一个数字列和一个文本列，而 val_1 指的是列 text_1 下任意一个单元格的值。

给定一个 SQL 模板，在每次实例化时，TAPEX 会均匀地从采样到的表格的列和单元格值抽取合理的部分来填充 SQL 模板，从而构成一个具体的 SQL 语句。值得注意的是，执行结果为空的 SQL 查询语句会被丢弃，因为空的结果无法很好地反映程序执行的逻辑。通过这种方式，TAPEX 可以大规模地构造高质量的预训练语料库。同时，TAPEX 还可以自由地控制预训练语料库的规模，因为理论上可以合成的预训练语料是无限多的。本文实验中 TAPEX 使用的最大的预训练语料包括了 5,000,000 个 SQL 查询语句和它们的执行结果。在下文中，除非明确说明，否则所有的实验结果都默认为在最大的预训练语料下进行预训练得到的结果。

4.5　实验与验证

在本节中，三个知名的弱监督语义解析数据集上的实验验证了本章提出生成式弱监督语义解析方法的有效性，以及执行引导的预训练方法的优越性。

4.5.1　实验设置

表 14　实验用数据集上的模型输入输出样例

数据集	模型输入示例	模型输出示例
WikiSQL	How many CFL teams are from York College? [HEAD] : pick # | CFL team Player | Position | College [ROW] 1 : 27 | hamilton tiger-cats | connor healey | db | wilfrid laurier [ROW] 2 : 28 | calgary stampeders | anthony forgione | ol | york . . .	2
	问题翻译：有多少 *CFL* 球队来自约克学院？	答案翻译：*2*
WikiTableQuestions	Which album released by the band schnell fenster produced the most singles appearing on the Australian peak chart? [HEAD] : Year | Title | Peak Chart Positions AUS | Peak Chart Positions NZ | Album [ROW] 1 : 1988 | "whisper" | 58 | 42 | the sound of trees [ROW] 2 : 1988 | "love-hate relationship" | 81 | 46 | The Sound Of Trees . . .	The Sound Of Trees
	问题翻译：哪一张 *schnell fenster* 乐队发行的专辑入选澳大利亚金曲榜单的次数最多？	答案翻译：树木之声
SQA	Where are the players from? Which player went to Louisiana State University? [HEAD] : Pick | Player | Team | Position | School [ROW] 1 : 1 | Ben McDonald | Baltimore Orioles | RHP | Louisiana State University [ROW] 2 : Tyler Houston | Atlanta Braves | C | Valley HS (Las Vegas, NV) . . .	Ben McDonald
	问题翻译：运动员们都来自哪？谁上过路易斯安那州立大学？	答案翻译：本-麦克唐纳

(1) 数据集与评价指标

本章选择了三个经典的弱监督语义解析数据集 WikiSQL, WikiTableQuestions 和 SQA 作为实验的数据基准，其中 WikiSQL 和 WikiTableQuestions 是以句子为单位构建的，而 SQA 则是一个对话为单位的基准数据集。从模型学习的难度来看，WikiSQL 和

表 15　数据集规模统计

数据集	类型	问题数量	表格数量
WikiSQL	简单问答	87,673	24,241
WikiTableQuestions	复杂问答	22,033	2,108
SQA	简单对话问答	17,553	982

SQA 相对简单，而 WikiTableQuestions 相对困难。与只需要在表格单元格上进行过滤（Filter）和聚合（Aggregate）的 WikiSQL 数据集相比，WikiTableQuestions 数据集需要模型具有复杂的推理能力，例如对给定的表格进行排序（Order）。表 14展示了三个数据集上的模型输入输出示例，各个数据集的规模统计如表 15所示。

对于所有数据集，模型的评价指标都是**答案准确率**（Denotation Accuracy）。对每个样本而言，当模型预测的答案与人类标注的答案完全一样时，答案准确率为 100%，否则答案准确率为 0%。

(2) 实现细节

本章所提出的方法是基于 fairseq[138] 实现的。在模型预训练期间，模型最多会被优化 50,000 次，每次优化会使用 256 个预训练样本上反向传播得到的平均梯度，学习率被设置为 3×10^{-5}。在 8 块 Tesla V100 GPU 上，大概仅需要 36 小时即可完成一次预训练。在模型微调期间，模型最多会被优化 20,000 次，每次优化使用 128 个下游数据集的样本，学习率与预训练时保持一致。在 SQA 相关的实验中，由于本章的重点并非语境建模，因此模型直接采取最简单的方法处理对话语境，即将对话的历史轮与当前轮自然语言拼接作为模型输入，如表 14所示。

4.5.2　实验结果

表 16展示了不同模型在 WikiSQL 上的性能，其中 BART 和 TaPEx 分别代表了本章所实现的基于 BART 模型的生成式弱监督语义解析方法和 BART 模型通过继续预训练强化后的对应方法。值得注意的是，表格中报告的是本章所提方法 5 次随机试验的平均性能，对表 17和表 18也是如此。正如表中所展示的，即使 BART 之前并没有在弱监督语义解析相关数据上进行过领域相关的预训练，但它单独就可以在 WikiSQL 的测试集上取得高达 85.8% 的答案准确率，表明生成式语言模型在下游数据集数据量较丰富时已经可以打败前人的弱监督语义解析方法，验证了用生成式语言模型做弱监督语义解

表 16　不同模型在 WikiSQL 数据集上的答案准确率，带有 ♡ 的模型在训练时使用了程序标注

模型	开发集	测试集
MAPO [64]	71.8	72.4
MeRL [65]	74.9	74.8
Abstract Program [129]	79.4	79.3
Discrete Hard EM [68]	84.4	83.9
TaPaS [44]	85.1	83.6
GraPPa [46]	85.9	84.7
Discrete Hard EM + Execution-Guided Decoding [139]	87.4	87.2
BART（本方法）	87.3	85.8
TaPEx （本方法）	**89.2**	**89.5**
SeaD♡ [140]	90.2	90.1

析任务的有效性。同时，在被 TaPEx 增强后，本方法以极大的优势超过了所有基线模型。例如，在 WikiSQL 的测试集上，TaPEx 达到了 89.5% 的答案准确率，比基线模型中的最佳性能还要高 2.3%。注意到最佳的基线模型（Discrete Hard EM + Execution-Guided Decoding，离散采样最大化期望算法 + 执行引导的解码）在推理时也同样使用了 SQL 执行器，TaPEx 能取得这样的优势是非常不易的。最终，本方法在著名的弱监督语义解析基准数据集 WikiSQL 上取得了最先进的结果，并与程序标注下最先进的基线模型 SeaD 取得了可比的性能，实现了弱监督下答案驱动的语义解析模型的性能与程序标注训练模型性能可比的目标。

如表 17所示，在更具挑战性的基准数据集 WikiTableQuestions 上，本方法也取得了高达 57.5% 的答案准确率，比之前最好的系统超出了 4.8%。但同时，注意到单独的 BART 模型只能达到 38.0% 的答案准确率，比其他用到预训练语言模型的基线模型要差很多。这可能是因为 WikiTableQuestions 的训练数据量相对较小，从较小规模的训练数据学习弱监督语义解析相关的归纳偏置对 BART 来说较为困难。不过，执行引导的预训练方法 TaPEx 的加入可以让 BART 有 19.5% 的大幅改进，这表明 TaPEx 可以显著减少 BART 对下游数据集训练样本数量的需求。与程序标注下最先进的基线模型 SDCUP 相比，本方法将强弱监督中最好模型的差距缩小到了 2.1%，令人印象深刻。

表 18展示了不同模型在 SQA 测试集上的表现，其中 ALL 是所有句子平均的答案准确率，SEQ 是所有对话平均的答案准确率，$\mathbf{Q}_i$ 是对话中第 i 轮的平均答案准确率。从表格中可以看出，TaPEx 不论是在对话级别（48.4%）还是句子级别（74.5%）上都获得了最先进的答案准确率。这样的提升并非易事，因为 SQA 是一个以对话为单位的弱监

表 17　不同模型在 WikiTableQuestions 数据集上的答案准确率，带有 ♡ 的模型在训练时使用了程序标注

模型	开发集	测试集
Float Parser [62]	37.0	37.1
Neural Programmer [141]	34.1	34.2
Marco Grammar [142]	40.6	43.7
MAPO [64]	42.7	43.8
Iterative Search [128]	43.1	44.3
MeRL [65]	43.2	44.1
Abstract Program [129]	43.7	44.5
TaPaS [44]	–	48.8
TaBERT [45]	53.0	52.3
GraPPa [46]	51.9	52.7
BART（本方法）	37.2	38.0
TaPEx （本方法）	**57.0**	**57.5**
SDCUP♡ [143]	–	59.6

督语义解析数据集，而 TaPEx 是以单个 SQL 语句为单位进行预训练的，并没有涉及上下文的理解。同时，与 BART 相比，TaPEx 在 SQA 上的大幅改进也验证了表 17上的结论，即 TaPEx 可以极大地缓解低资源下的弱监督语义解析问题。

正如在4.3.2节中所介绍的，生成式弱监督语义解析因为使用同样的架构处理各种下游弱监督语义解析任务，可以很容易地进行多任务学习。为了验证这一点，表 19中展示了多任务学习的实验结果，其中**源数据集** $\mapsto$ **目标数据集**的含义是指模型首先在源数据集上微调，再在目标数据集上微调。从表格中不难得到如下结论：

1、当 BART 作为预训练模型时，多任务训练能显著提升模型在目标数据集上的性能。

2、当 TaPEx 作为预训练模型时，多任务训练的好处不太明显，这表明通过多任务学习可迁移的能力大部分可以通过本章所提出的预训练方法获得。

4.5.3　实验分析

(1) 预训练的性能分析

TaPEx 采用的预训练任务是预测给定 SQL 语句的执行结果。为了探索模型预训练后在该任务上的性能，本节分析了 TaPEx 在近 20,000 个从未见过的 SQL 查询语句上

表 18　不同模型在 SQA 测试集上的答案准确率

模型	ALL	SEQ	Q_1	Q_2	Q_3
Float Parser [62]	33.2	7.7	51.4	22.2	22.3
Neural Programmer [69]	40.2	11.8	60.0	35.9	25.5
DynSP [144]	44.7	12.8	70.4	41.1	23.6
CAMP [145]	45.6	13.2	70.3	42.6	24.8
GNN [146]	55.1	28.1	67.2	52.7	46.8
SCoRe [147]	65.4	38.5	78.4	65.3	55.1
TaPaS [44]	67.2	40.4	78.2	66.0	59.7
TaPaS + Intermdiate [148]	71.0	44.8	**80.9**	70.6	64.0
BART（本方法）	58.6	27.8	65.3	54.1	57.0
TaPEx （本方法）	**74.5**	**48.4**	76.2	**71.9**	**76.9**

表 19　在目标数据集上多任务训练的答案准确率结果

源数据集 ↦ 目标数据集	BART	TaPEx
WikiSQL ↦ WikiTableQuestions	47.4	57.2
WikiTableQuestions	37.2	57.0
WikiSQL ↦ SQA	64.1	70.8
SQA	57.5	70.3

的性能。从结果来看，TaPEx 在完成 SQL 执行任务上的性能较好，因为 89.6% 的 SQL 查询语句的执行结果都可以被模型正确预测。特别是，TaPEx 在过滤（Filter），聚合（Aggregate）和极值（Superlative）相关操作上的表现更好，表明它在表格的单元选择和聚合计算上的准确性很高。至于算术（Arithmetic）和比较（Comparative）操作，TaPEx 也做得很好，显示了它在弱监督语义解析场景下强大的数值推理能力。总而言之，TaPEx 是一个学会了表格选择、表格聚合和数值推理能力的神经网络化的 SQL 执行器。

(2) 下游任务性能的细粒度分析

为了理解预训练具体在哪些问题上会对模型有明显的帮助，本节从 WikiTableQuestions 开发集上随机挑选了 500 个问题，对它们进行了分析归类，并在表 20中报告了不同类别的问题上 BART 和 TaPEx 的答案准确率。分析结果显示，相比于 BART，TaPEx 在所有问题类别上性能都有明显提升，这意味着它全方位地增强了模型对文本和表格进行联合推理的能力。

表 20　常见的 7 种语义操作，示例问题以及模型在各种问题上的答案准确率

问题类别	示例问题	BART	TAPEX
选择（Select）	What is **the years won** for each team? 翻译：每支队伍分别是在哪年赢球的？	41.3%	64.8% (+23.5%)
过滤（Filter）	How long did **Taiki Tsuchiya** last? 翻译：*Taiki Tsuchiya* 活了多久？	40.1%	65.7% (+25.6%)
聚合（Aggregate）	What is the **amount of** matches drawn? 翻译：有多少场比赛？	26.9 %	57.4% (+30.5%)
极值（Superlative）	What was the **last** Baekje Temple? 翻译：哪个是最后一个百济寺？	46.3 %	64.3% (+18.0%)
算术（Arithmetic）	What is the **difference** between White voters and Black voters in 1948? 翻译：*1948* 年，白人选民和黑人选民的选票差了多少？	33.1 %	53.5% (+20.4%)
比较（Comparative）	Besides Tiger Woods, what other player won **between 2007 and 2009**? 翻译：除了泰格伍兹，还有哪位球员在 *2007* 年和 *2009* 年之间曾赢过比赛？	30.0 %	55.9% (+25.9%)
分组（Group）	What was score **for each** winning game? 翻译：每场获胜的比赛的得分是多少？	49.5 %	66.7% (+17.2%)

(3) 预训练规模对下游任务性能的影响分析

图 41展示了不同规模预训练语料对下游数据集性能的影响。从图中可以看出，即使 TAPEX 所使用的预训练语料库是自动合成的，扩大预训练语料库的规模一样会带来积极的影响。这个观察结果与前人工作利用自然语言做语言建模的结论类似[42]：预训练语料库越大，下游任务上的表现越好。通过对比不同数据集上的性能差异可以发现，对于像 WIKISQL 这样的简单任务，扩大预训练语料的收益变得微弱，而对像 WIKITABLEQUESTIONS 这样的复杂任务，扩大预训练语料规模带来的收益仍非常显著。同时，通过增加预训练语料的规模，模型在低资源的 WIKISQL 和 SQA 数据集上的答案准确率一直保持增长趋势。总结来讲，当下游任务难度较大，或者下游任务的训练样本数量较少时，TAPEX 的预训练语料的规模影响较大。

(4) 预训练方法的效率比较

正如在4.1节中提到的，现有的弱监督语义解析预训练方法的预训练效率相对较低，它们往往需要极其大的预训练语料库才能使下游任务受益。为了对比分析 TAPEX 与前

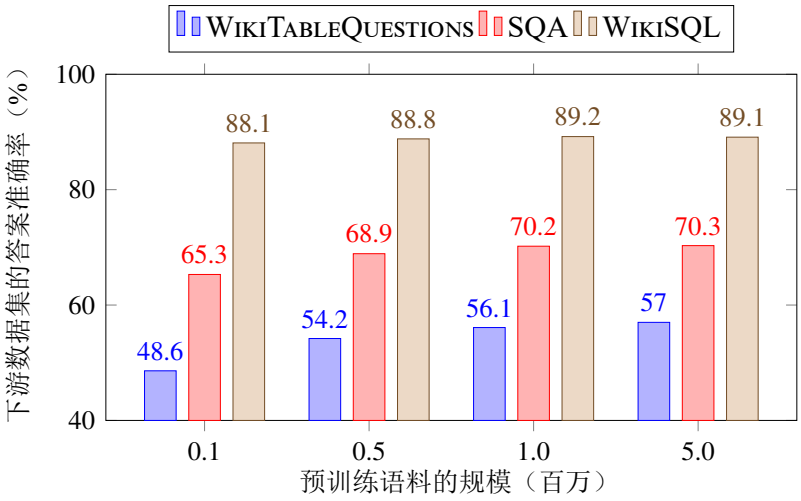

图 41　不同规模的预训练语料对下游数据集性能的影响

人工作预训练的效率，图 42中展示了不同预训练规模下 TaPEx 和基线预训练模型 TaPaS [44]、TaBERT [45]、GraPPa[46] 在 WikiTableQuestions 上的答案准确率。从图中可以分析得到，当仅使用 10 万的预训练数据，TaPEx 在 WikiTableQuestions 上依然有非常大的提升（+11.4%）。比起类似性能的 TaPaS 模型，TaPEx 的预训练相比 TaPaS 实现了 200 倍的加速比，证实了本章所提出的执行引导的预训练方法效率很高。

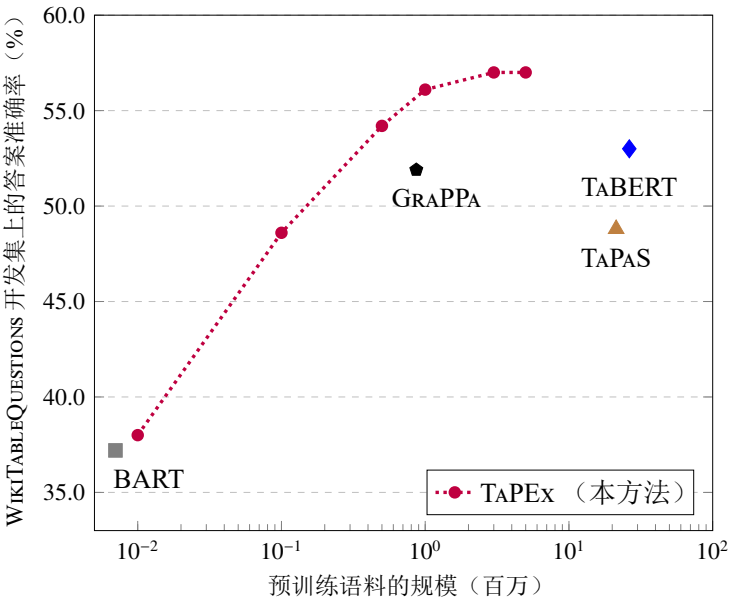

图 42　不同模型使用的预训练语料库规模（百万）与 WikiTableQuestions 开发集的答案准确率

(5) 自然语言作为预训练任务的输入

相比前人工作，本章所提出的预训练任务的一个特殊之处在于它的预训练输入是程序语言 SQL 查询语句，而非自然语言。直觉上说，与 SQL 查询语句相比，在预训练中使用自然语言对下游任务来说会更好，因为这样的话预训练任务与微调任务更加靠近。然而，要获得一个能忠实反映 SQL 查询语句的流畅的自然语言句子是不容易的。

表 21　SQL 语句、由模型翻译得到的自然语言句，以及该自然语言是否忠实反映 SQL 语句的语义

SQL 查询语句	翻译的自然语言句	忠实度
SELECT Name WHERE Age >= 28	Who is at least 28 years old? 翻译：谁 *28* 岁以上？	✓
SELECT MAX (Pick#)	What was the last pick in the 1989 major league baseball draft? 翻译：*1989* 年大联盟棒球选秀中的最后一个选手号是多少？	✗
SELECT Driver ORDER BY Pos DESC LIMIT 1	What driver came in last place? 翻译：哪位司机排在最后一名？	✓
SELECT COUNT (Competition) WHERE Notes != 100	How many competitions have no notes? 翻译：有多少比赛没有任何说明？	✗
SELECT COUNT (*) WHERE Result = “won” AND Year > 1987	How many times did they win after 1987? 翻译：*1987* 年后他们赢了多少次？	✓
SELECT MAX (Chart Position) - MIN (Chart Position) WHERE Release date = “july 21, 1995”	What is the difference between the chart position of july 21, 1995 and the chart position of july 22, 1995? 翻译：*1995* 年 *7* 月 *21* 日的排行榜与 *1995* 年 *7* 月 *22* 日的排行榜之间差多少？	✗
SELECT Nation WHERE Nation != “Japan” AND Gold = (SELECT Gold WHERE Nation = “Japan”)	Which other countries had the same number of gold medals as Japan? 翻译：哪些国家的金牌数量与日本相同？	✓
SELECT Incumbent Electoral History GROUP BY Incumbent Electoral History ORDER BY COUNT (*) DESC LIMIT 1	Who has held the office the most? 翻译：谁担任过最多的职务？	✗

参考前人工作[137]，本文训练了一个给定 SQL 语句，生成其对应自然语言的模型，并利用该模型将预训练语料库中的 SQL 查询语句全部翻译成了自然语言。该模型基于 BART 构建，并在 SQUALL 数据集[115] 提供的约 9000 个 SQL 语句与自然语言句的成对数

表 22　TAPEX-SQL 和 TAPEX-自然语言在下游任务数据集上微调的答案准确率结果对比

设定	TAPEX-SQL	TAPEX-自然语言
WIKISQL	88.8	87.5
WIKITABLEQUESTIONS	54.2	52.8
SQA	68.9	68.7

据上进行训练。将训练好的模型应用于 50 万训练样本的预训练语料库，可以获得一个相同规模的由自然语言构成的预训练语料库。通过在 100 个随机抽样的 SQL 语句分析后可以发现：模型生成的自然语言几乎都很流畅，但只有 68% 的自然语言句能较好地反映对应 SQL 语句的语义。表 21展示了一些随机抽样的 SQL 语句和它们相应的翻译的自然语言句。

在获得了自然语言预训练语料库之后，可以使用 TAPEX 相同的预训练和微调程序来得到该预训练目标在下游任务上的性能。为方便区分，下文中原始的 TAPEX 被称为 TAPEX-SQL，而在预训练任务中使用自然语言预训练的模型被称为 TAPEX-自然语言。表 22中展示了这两个预训练模型在下游任务上的性能对比。从表格中可以看出，非常令人惊讶地，TAPEX-自然语言的效果与 TAPEX-SQL 相比要更差一些。例如，与在预训练中使用 SQL 语句相比，使用自然语言会导致预训练模型在 WIKITABLEQUESTIONS 开发集上的答案准确率下降 1.4%。经过仔细分析，这种程度的下降可能是因为模型所翻译的自然语言中包含了一些噪音。以表 21中第二行的数据为例，自然语言句中的信息 “1989 major league baseball draft” 在 SQL 语句中并无体现，而这种语义上的噪音可能会干扰预训练的性能。

4.6　本章小结

在本章中，为提升语义解析模型在弱监督下的性能，本章首先提出了一种弱监督语义解析的新思路，该思路利用生成式模型同时发挥语义解析模型和程序执行器的功能，直接可微地解决弱监督语义解析任务。在该思路的基础上，本章提出了一种执行引导的预训练方法以继续提升生成式语言模型在弱监督语义解析场景上的性能。与一般预训练使用大规模的自然语言语料库不同，该预训练方法的预训练语料是通过随机采样 SQL 查询语句从而自动合成的。通过在多样化、大规模和高质量的合成语料库上学习复杂 SQL 语句的执行过程，该预训练方法隐式地强化了生成式模型中的程序执行器。在三个不同的弱监督语义解析数据集上的实验结果表明，本章所提出的使用生成式模型解决弱监督语义解析任务是有效的。同时，执行引导的预训练方法可以继续提升生成式模型的

性能，尤其在低资源场景中，相比原始生成式模型最高可以取得 19.5% 的绝对提升。实验分析表明，与前人的预训练方法相比，本章所提出的预训练方法可以取得将近 200 倍的加速比，预训练效率非常高。最终，本方法在所有数据集上都取得最先进的结果，在两个数据集上性能甚至可与强监督程序标注下最好的基线模型持平。

局限性与未来工作 本方法因为利用语言模型接受整张表格作为输入，因此在处理比较大型的表格时较为困难。当用户端的表格相对较小时，序列化后的表格长度仍在预训练语言模型允许的位置编码范围内，此时本方法的效果比较好。但当用户端的表格太大（如超过 500 个单元格）时，模型需要的显存会随着表格长度的增加而平方倍增长，以至于已有的显卡无法容纳得下，此时本方法无法接受整张表格作为输入。在实践时，本方法通过删除一些不相关的行或列来压缩表格的大小，但这样的压缩会降低在相应任务上的模型性能。因此，一个重要的未来工作是改进本方法使其可以适用于更大的表格。

第五章　半监督下对话重写驱动的对话式自然语言语义解析方法

针对对话标注构造难的难点，为提升对话式语义解析模型在半监督下的性能，本章提出将对话式语义解析任务解耦为对话重写和单轮语义解析，并提出基于编辑的对话重写方法，使得半监督下对话式语义解析模型可达到全监督训练 65% 的性能。

5.1　引言

年份	城市	国家	参赛国家数量
1896	雅典	希腊	14
1900	巴黎	法国	24
1904	路易斯	美国	12
…	…	…	…
2004	雅典	希腊	201
2008	北京	中国	204
2012	伦敦	英国	204

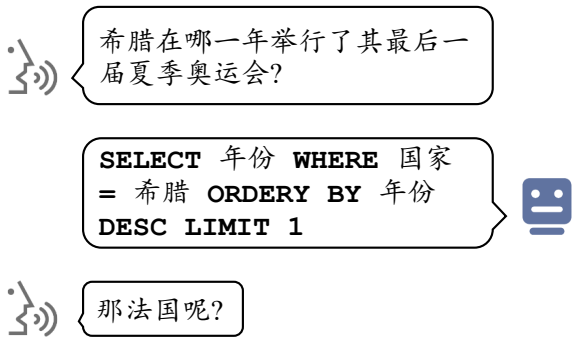

图 43　对话式语义解析场景示意图

语义解析能够减轻用户学习编程语言的负担，允许没有编程背景的用户通过自然语言零门槛地完成本需编程才能完成的任务。前人工作主要关注在对一句语境无关自然语言的解析，但在人机互动中用户往往会问一连串问题，推动了对话式语义解析的相关研究。如图 43所示，在这种场景中用户会期望模型不仅能理解语境无关的问句“希腊在哪一年举行了其最后一届夏季奥运会?”，也能理解语义依赖于语境的问句“那法国呢?”。可以看出，相比于在单轮语义解析中用户必须每次都给出完整的自然语言如“法国在哪一年举行了其最后一届夏季奥运会?”，对话式语义解析场景允许用户的问句中带有指代（Coreference）和省略（Ellipsis），对用户来说更加自然和友好。同时，因为它允许模型与用户进行多轮交互，对话式语义解析也是未来重要的人机交互接口。

然而，相比于受到极大关注的单轮语义解析[2]，对话式语义解析却鲜少受到关注，从相关数据集的发展就可看出趋势。语义解析的数据集目前已遍地开花，实现了跨领域[10]、跨语言[149]、跨应用场景[150]。相比之下，对话式语义解析的工作仍主要关注在单领域数据集 ATIS3[70] 和较为简单的数据集 SQA[144]。最近，研究者们也提出了几个跨领域的语义解析数据集，如英文数据集 SParC[71] 和中文数据集 CHASE[73]，但这些数据集最多的也仅包含约 5,000 个对话，比目前规模最大的单轮语义解析数据集 WikiSQL[11] 的规模小一个数量级。高质量数据集的缺失并非偶然，相比于单轮语义解析，自然流畅的

对话式语义解析数据集的收集要更加昂贵与困难。正如 Guo 等[73] 所指出的，不当的对话式语义解析数据集构造方式会让数据分布有所偏差。例如，作为一个较受认可的跨域对话式语义解析数据集，SParC 中却有将近 48% 的问题都是语境无关的，这与前人工作在真实场景所统计的比例相去甚远[151]。以上种种现状表明，对话标注构造难是对话式语义解析模型遇到的主要难题。

对比之下，单轮语义解析已经有较多的高质量训练数据。如果让单轮语义解析已有的数据集可以惠及对话式语义解析，构造难的问题就可以迎刃而解。实际上，对于对话式语义解析来说，让它有别于单轮语义解析的主要特点在于**对话中是否有语义依赖于语境的语句**。如果将对话中依赖于语境的语义被补充完整，使得一个单轮语义解析器也可以轻松地理解它们，单轮语义解析的数据就可以直接利用于对话式语义解析。

针对对话标注构造难的难点，本章提出将对话式语义解析任务解耦成**单轮语义解析**和**对话重写**两个子任务的方法。单轮语义解析即前人工作主要关注的语义解析方向，旨在于将语境无关的语句解析成对应的计算机程序并完成复杂任务，而对话重写的目标是将对话中每一个语境相关的自然语言改写成一个语义等价但语境无关的自然语言。以图 43中的问句 “那法国呢?” 为例，对话重写任务的目标是将其重写为 “法国在哪一年举行了其最后一届夏季奥运会?”。通过标注成本低廉的对话重写数据，现有的单轮语义解析数据可以被驱动从而实现半监督[1]下的对话式语义解析。

表 23　开放域多轮对话重写样例，样例一和样例二是指代和省略的典型场景[152]

	样例一
第一轮：	梅西有多高？
第二轮：	官方说法他的身高是 5 英尺 7 英寸。
第三轮：	他和 C 罗谁是最好的球员？
第三轮重写句：	梅西和 C 罗谁是最好的球员？
	样例二
第一轮：	你最喜欢什么电影？
第二轮：	泰坦尼克。
第三轮：	为什么呢？
第三轮重写句：	为什么最喜欢泰坦尼克？

实际上，在开放域对话，前人工作已经证实了对话重写对于降低数据标注代价的积极意义。2019 年，Su 等[152] 首次提出开放域对话系统的对话重写任务，并构建了一个

[1]这里半监督指的是面向对话式语义解析，本方法仅需要对话中首句自然语言对应的语义解析监督，而不需要其它轮自然语言对应的语义解析监督数据。

大规模的开放域对话重写数据集，其中两个样例如表 23所示。类似地，CANARD 数据集[153] 提出了对话式阅读理解中对话重写任务的重要性。然而，对话式语义解析还尚无相关工作通过引入对话重写来降低对话式语义解析数据标注的难度。因此，本章首先构建了首个面向语义解析的对话重写数据集 **FollowUp**，它包含了 1,000 个对话，覆盖了 120 张不同领域的表格。由于对话重写的目标是生成对话，于是本章在所构建的数据集上使用最先进的序列生成方法，如序列到序列模型[26] 作为基线模型进行了实验。实验结果表明，对话重写任务作为单轮语义解析和对话式语义解析的桥梁，可以将一个单轮语义解析器扩展到对话式语义解析场景，并发挥模型 40% 的能力。然而，由于 FollowUp 数据集的数据多样，序列生成方法在对话重写任务上性能不够好，是影响对话式语义解析性能的主要瓶颈。受重写后的语句大部分单词均来自原始对话特点的启发，本章提出了一个新颖的对话重写模型 STAR（Split-And-Recombine，拆分后重组）。FollowUp 数据集上的实验结果清楚地表明，STAR 模型能显著改善对话重写的效果，从而提升对话式语义解析的性能。最终，STAR 可以让一个单轮语义解析器在对话式语义解析场景下发挥高达 65% 的性能。

5.2 总体结构

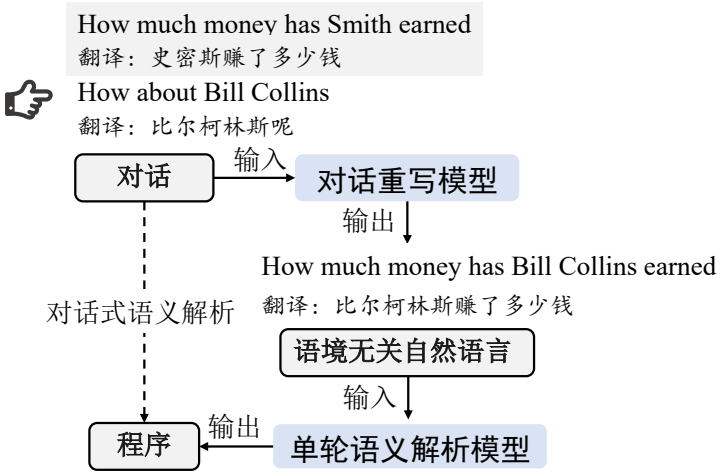

图 44　本章方法示意图：引入对话重写与单轮语义解析一起完成对话式语义解析任务

不同于前人工作端到端地完成对话式语义解析，本章所提的方法包含两个独立训练的部分（如图 44所示）：首先，一个训练好的对话重写模型以整个对话为输入，将当前语义依赖于语境的自然语言（如图中的 “How about Bill Collins”）重写为一个语境无关、但语义不变的自然语言（如 “How much money has Bill Collins earned”）；接着，一个在单轮语义解析数据集上训练好的单轮语义解析模型可以直接理解该语境无关的自然语言，并生成它对应的程序。可以看出，本章所提出的方法将对话式语义解析解耦为两个独立

的任务，可以很好地将已有的单轮语义解析模型扩展完成对话式语义解析的功能。

5.3　面向对话重写的数据集构建

本节介绍数据集的标注与数据集所采用的评价指标。

5.3.1　数据集标注

FollowUp 数据集包括了 1,000 个通过众包标注的对话，这些对话覆盖了 120 个不同的表格，并涵盖了多个典型的对话场景[2]。根据 Bertomeu 等[151] 的研究，对话中有约 75% 的语境依赖是依赖于上一句，因此 FollowUp 数据集中的每个对话主要由**语境句**和**当前句**两句话构成。其中语境句是一句语义完整的自然语言，而当前句是可能包含指代现象或省略现象的自然语言。对话重写的目标是将每个当前句改写成一个语义独立于上下文的**重写句**，也即数据集需要标注的对话重写的监督信息。数据集中涵盖的 8 个主要场景和对应的语境句、当前句和重写句的示例如表 24所示。

不同于语义解析需要懂程序的领域专家，普通人就可以胜任对话重写任务的标注。因此，FollowUp 数据集由 8 位熟练使用英文的普通人标注构造。为控制数据质量，数据标注的流程主要分为三个阶段：

1、表格选择：在维基百科上筛选后选择 120 张高质量的表格，每张表格至少有 8 行和 1 列由数字构成的列。

2、语境句构造：标注员首先会在每个表格上构造一些语义完整的语境句。为了增加数据集的多样性，标注过程会随机显示一些语义提示语，如 “需要显式地说明排序”[62]。

3、对话构造：在语境句标注完成后，标注员会以随机展示的语境句作为背景，写出语义可能依赖于该语境句的当前句和其对应的重写句。对于每种对话重写场景，数据标注时都有 10 个例子供标注员参考，从而保证整个数据集的多样性。

在完成数据集的构造后，标注员还需要交叉验证所标注的重写句是否表达了与当前句一样的语义，语义不一致的标注样本将被丢弃。

5.3.2　评价指标设计

合理的评测指标是公平比较数据集上不同模型性能的必需条件。考虑到对话重写的主要目标是生成重写句，本节引入了 3 个不同的评价指标来比较模型预测的重写句和标注的重写句之间的匹配程度：

BLEU 由于对话重写可以看作是翻译任务的一个特例，因此数据集引入了机器翻

[2]数据集已开源在http://github.com/SivilTaram/FollowUp

表 24　8 个典型的对话重写场景和对应样例

场景	样例
选择	语境句：In 1995, is there any network named CBC? 翻译　：在 *1995* 年，是否有一个名为 *CBC* 的网络品牌？ 当前句：Any TSN? 翻译　：有叫 *TSN* 的吗？ 重写句：In 1995, is there any network named TSN? 翻译　：在 *1995* 年，是否有一个名为 *TSN* 的网络品牌？
比较	语境句：How much money has Smith earned? 翻译　：史密斯赚了多少钱*?* 当前句：Compare it with Bill Collins. 翻译　：与比尔柯林斯相比。 重写句：Compare money Smith earned with Bill Collins. 翻译　：比较史密斯与比尔柯林斯赚的钱。
运算	语境句：List all universities founded before 1855. 翻译　：列出所有 *1855* 年以前成立的大学。 当前句：Show their number. 翻译　：有多少？ 重写句：Show the number of all universities founded before 1855. 翻译　：有多少大学是 *1855* 年以前成立的呢？
极值	语境句：Which stadium has the most capacity? 翻译　：哪座体育场的容量最大？ 当前句：Which get the highest attendance? 翻译　：哪座的观众出席率最高？ 重写句：Which stadium get the highest attendance? 翻译　：哪座体育场的观众出席率最高？
过滤	语境句：How many roles are from studio paramount? 翻译　：有多少角色来自派拉蒙工作室？ 当前句：List all titles produced by that studio. 翻译　：列出该工作室出品的所有作品。 重写句：List all titles produced by studio paramount. 翻译　：列出派拉蒙工作室出品的所有作品。
分组	语境句：Show the industry which has the most companies? 翻译　：显示公司数量最多的行业。 当前句：Show in different countries. 翻译　：加上国家信息。 重写句：Show the industry which has the most companies in different countries. 翻译　：显示在不同国家公司数量最多的行业。
排序	语境句：Show all chassis produced after the year 1990. 翻译　：显示 *1990* 年以后生产的所有底盘。 当前句：Sort them by year. 翻译　：按年排序。 重写句：Show all chassis produced after the year 1990 and sort by year. 翻译　：显示 *1990* 年以后生产的所有底盘并按年排序。
搜索	语境句：What position did Sid O'Neill play? 翻译　：希德奥尼尔打什么位置？ 当前句：Which players else are in the same position? 翻译　：还有哪个运动员打同一位置？ 重写句：Which players play in the position of Sid O'Neill excluding Sid O'Neill? 翻译　：除了希德奥尼尔外，还有哪个运动员和他打同一位置？

表 25　关键词表格

关键词类别	示例
表格列名	销售额, 国家
表格单元格值	2018, 澳大利亚
汇总词	总和，最大，数量
比较词	更多，之后
排序词	下降，上升

译中的经典指标 BLEU[154] 作为一个评价指标。该指标会根据每个预测重写句和标注重写句之间基于 $N(1 \le N \le 4)$ 元词段重合率的均值作为它们之间的相似度。

关键词保留率 BLEU 指标对每个单词都一视同仁，然而从语义解析器的角度讲某些单词（如表格的列名）比其他单词更加重要。因此，一个脚本被构造以用于自动抽取重写句中的关键信息，一些示例如表 25所示。对于每个预测的重写句和标注的重写句，该脚本都会抽取出它们中所包含的关键信息，并根据预测重写句中关键信息占标注关键信息的比例来计算关键词保留率。下文统一使用 SymAcc 指代该指标。

执行准确率 为了量化地评估预测的重写句对于对话式语义解析任务的作用，本章使用先进的语义解析器解析每个重写句，并通过执行该重写句对应得到的 SQL 语句得到的答案与标注答案是否一致来评估该重写句的执行准确率。具体地，本章使用 WikiSQL 数据集[11] 上最先进的语义解析器 Coarse2Fine[28] 来得到每个重写句对应的 SQL 语句。下文统一使用 ExAcc 指代该指标。

5.4　基于拆分重组的对话重写

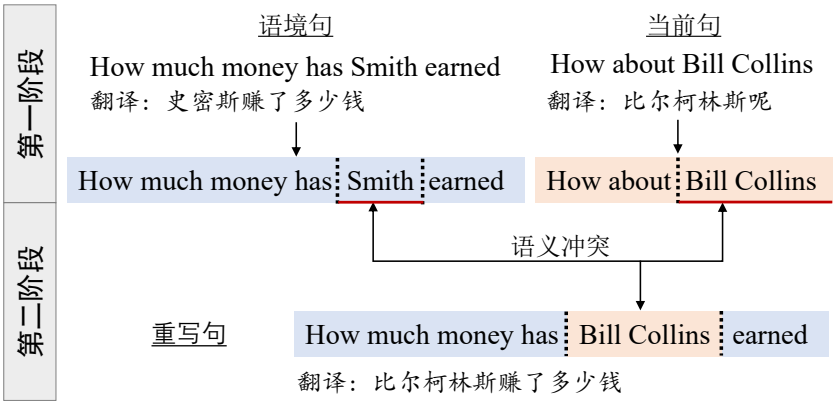

图 45　片段作为最小单位以拆分和重组输入的语境句和当前句

不同于直接逐词解码生成重写句的序列生成模型，本节提出一个新颖的基于拆分和重组的 STAR 模型，该模型旨在通过对话的重组来构成重写句。如图 45所示，该模型首先引入片段（Span）作为粒度更适中的操作单位。给定一个对话，模型首先应用类似于序列标注（Sequence Labeling）的方法识别出每个句子中需要拆分的位置，产生了若干语义相对独立的片段。接着，模型逐个计算两两片段之间的相似度，相似度超过一定阈值的片段对被模型认为语义存在重复。在生成重写句时，模型在两个语义有重复的片段中只会保留新的那个，而舍弃旧的片段。通过这样的方式，基于拆分重组的模型可以得到一个充分利用对话信息构造的重组句。虽然模型在推理时需要以片段为单位进行编辑，训练数据中却并没有片段相关的监督信息，所以该模型无法使用监督学习的范式来训练。因此，本章将该问题建模成一个强化学习问题，模型通过在片段的候选空间进行采样来不断探索与靠近正确的解空间，从而完成模型训练的目标。

5.4.1 模型架构

形式化地，令 $\mathbf{x} = (x_1, \ldots, x_n)$，$\mathbf{y} = (y_1, \ldots, y_m)$ 和 $\mathbf{z} = (z_1, \ldots, z_l)$ 分别表示语境句，当前句和重写句，其中 n，m 和 l 分别代表句子中包含的单词数。STAR 模型 $P_{\text{model}}(\mathbf{z}|\mathbf{x}, \mathbf{y})$ 的目标是学习在语境 $\mathbf{x}$ 下理解 $\mathbf{y}$，并生成对应的重写句 $\mathbf{z}$。$\mathbf{z}$ 和 $\mathbf{y}$ 的语义一致，但 $\mathbf{z}$ 是语义完整且无歧义的，可以直接作为一个训练好的单轮语义解析器模型的输入。当给定 $(\mathbf{x}, \mathbf{y})$ 时，$P_{\text{model}}(\mathbf{z}|\mathbf{x}, \mathbf{y})$ 要最大化如下的学习目标 $\mathcal{L}$：

$$\mathcal{L} = \mathbb{E}_{(\mathbf{x},\mathbf{y},\mathbf{z})\sim\mathcal{D}}[\log P_{\text{model}}(\mathbf{z}|\mathbf{x}, \mathbf{y})], \tag{5.1}$$

其中 $\mathcal{D}$ 表示训练数据。至于模型 $P_{\text{model}}(\mathbf{z}|\mathbf{x}, \mathbf{y})$ 的设计，如5.1节所述，考虑到 $\mathbf{z}$ 中大部分的词来自 $\mathbf{x}$ 和 $\mathbf{y}$，因此 P_{model} 被设计为一个 $\mathbf{x}$ 和 $\mathbf{y}$ 的合并过程。为此，STAR 模型引入了两阶段的过程，并结合强化学习来联合学习它们。

图 45中显示了在一个示例上两阶段的具体过程：

在第一阶段**拆分**中，语境句和当前句都被拆分若干个片段。例如，语境句被拆分为“How much money has”，“Smith” 和 “earned”。令 q 表示一种拆分 $(\mathbf{x}, \mathbf{y})$ 的方式，模型在第一阶段的目标被形式化为 $P_{\text{split}}(q|\mathbf{x}, \mathbf{y})$。

在第二阶段**重组**中，模型试图找出最可能的语义冲突对以重组片段，并通过重写过程产生最终的重写句，该过程可以形式化为 $P_{\text{rec}}(\mathbf{z}|q)$。需要说明的是，本章所说的语义冲突是指两个片段在语义上相似，比如 “Smith” 和 “Bill Collins” 都属于人名从而产生语义冲突。另外，任何语境句的片段和当前句的片段之间都有可能产生语义冲突。

模型 $P_{\text{model}}(\mathbf{z}|\mathbf{x},\mathbf{y})$ 的目标与这两个阶段的目标关系可用以下公式表示：

$$P_{\text{model}}(\mathbf{z}|\mathbf{x},\mathbf{y}) = \sum_{q\in Q} P_{\text{split}}(q|\mathbf{x},\mathbf{y}) P_{\text{rec}}(\mathbf{z}|q), \tag{5.2}$$

其中 Q 表示所有可能的拆分 $(\mathbf{x},\mathbf{y})$ 的方式。由于数据集中并没有拆分和重组对应的片段级别的监督数据，需要采用强化学习的方法来优化 P_{model}。令 $\tilde{\mathbf{z}}$ 表示模型预测的重写句，同时出于简洁考虑将 $\mathbb{E}_{(\mathbf{x},\mathbf{y},\mathbf{z})\sim\mathcal{D}}$ 简化为 $\mathbb{E}$，强化学习的目标即最大化函数 $\mathcal{L}_{\text{rl}}$：

$$\mathcal{L}_{\text{rl}} = \mathbb{E}[\sum_{\tilde{\mathbf{z}}\in\mathcal{Z}}\sum_{q\in Q} P_{\text{split}}(q|\mathbf{x},\mathbf{y}) P_{\text{rec}}(\tilde{\mathbf{z}}|q) r(\mathbf{z},\tilde{\mathbf{z}})], \tag{5.3}$$

其中 $\mathcal{Z}$ 是所有候选重写句的空间，r 代表强化学习中的奖励函数。

Q 和 $\mathcal{Z}$ 的笛卡尔乘积 $Q\times\mathcal{Z}$ 构成了模型输出的所有候选空间。然而，该输出候选空间非常大，使得较难直接优化 $\mathcal{L}_{\text{rl}}$。若直接采用 REINFORCE 算法[96]，即同时对 q 和 $\tilde{\mathbf{z}}$ 进行采样，采样效率较低，模型学习较为困难。为了缓解该问题，本节提出一种更有利于模型训练的采样方法，即首先对 q 进行采样，在 q 确定后枚举所有候选 $\tilde{\mathbf{z}}$。虽然相比直接的采样方法增加了计算成本，但该采样方法可以大幅度地压缩模型的候选空间，从而稳定模型的训练。因此，原始问题转变为设计一个用于评估采样到的 q 的质量的奖励函数 $R(q,\mathbf{z})$，从而指导模型的优化方向。为了实现该过程，公式 (5.3)中的目标函数 $\mathcal{L}_{\text{rl}}$ 可以使用如下公式改写：

$$\mathcal{L}_{\text{rl}} = \mathbb{E}[\sum_{q\in Q} P_{\text{split}}(q|\mathbf{x},\mathbf{y}) \sum_{\tilde{\mathbf{z}}\in\mathcal{Z}} P_{\text{rec}}(\tilde{\mathbf{z}}|q) r(\mathbf{z},\tilde{\mathbf{z}})]. \tag{5.4}$$

这样一来，仅需将 $R(q,\mathbf{z})$ 设为：

$$R(q,\mathbf{z}) = \sum_{\tilde{\mathbf{z}}\in\mathcal{Z}} P_{\text{rec}}(\tilde{\mathbf{z}}|q) r(\mathbf{z},\tilde{\mathbf{z}}), \tag{5.5}$$

即可达成模型优化的目标。

图 46展示了 StAR 模型的示意图。给定 $\mathbf{x},\mathbf{y}$，在第一阶段（图中蓝色部分），模型会固定 P_{rec} 的权重以提供奖励函数 $R(q,\mathbf{z})$，此时 P_{split} 可以通过 REINFORCE 算法学习。在第二阶段（图中红色部分），模型则固定 P_{split} 的权重，并通过它推理产生 q，此时 P_{rec} 可以被优化以最大化公式 (5.5)。通过这种方法，P_{split} 和 P_{rec} 可以被交替训练，下面将详细介绍这两个阶段的训练过程。

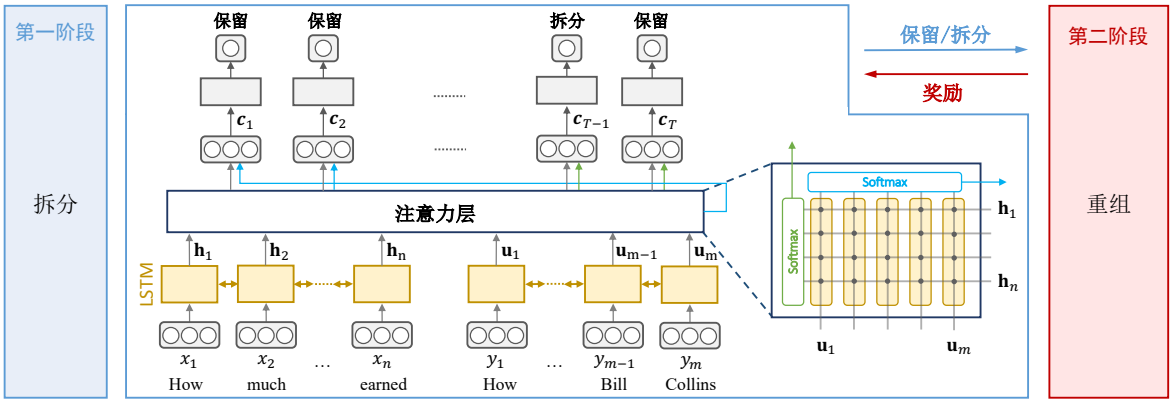

图 46　STAR 模型示意图：拆分和重组两个阶段通过强化学习共同优化

5.4.2　拆分阶段

如前文所述，拆分阶段的训练是由固定 P_{rec} 与更新 P_{split} 完成的。为方便描述，下文使用 SplitNet（Split Neural Network，拆分模型）来指代 P_{split}。如图 46所示，将语境句和当前句作为模型的输入，片段的拆分可以被建模成在输入的自然语言上进行序列标注。具体地，对于每个单词，SplitNet 都会输出一个二分类的决策结果**拆分**或**保留**来指示在相应单词后是否执行拆分操作。在每个单词上的决策结果最终构成了一个决策序列，而这个决策序列唯一地指定了一种拆分 $(\mathbf{x}, \mathbf{y})$ 的方式，对应前文中所说的 q。以图 46中的输入为例，因为语境句中的 “has” 和 “Smith” 词对应的决策结果均为拆分，因此模型会在它们后执行拆分操作，产生 “How much money has”，“Smith” 和 “earned” 三个片段。

(1) 拆分模型架构

直觉来说，只有在得到语境句和当前句交互的信息流，SplitNet 才能知道拆分这两句话的合理方式。受 BiDAF（Bi-Directional Attention Flow，双向注意力流）模型[155] 的启发，为了捕捉两句话之间的交互信息流，SplitNet 被设计为一个带注意力层（Attention Layer）的模型[27]。

在嵌入层，SplitNet 考虑了三个层次的嵌入，分别是**字符级**、**单词级**和**语句级**，它们分别用 ϕ_c、ϕ_w 和 ϕ_s 来表示。字符级的嵌入层采用卷积神经网络[156] 将每个单词映射到一个高维向量。单词级的嵌入层为每个单词维护一个高维向量，该向量使用 Glove 初始化[157]，并跟随其它模型参数一起优化。语句级的嵌入层是一个独热编码，用于区分语境句和当前句。总结下来，嵌入函数可以表示为 $\phi = [\phi_c; \phi_w; \phi_s]$。

嵌入层后，双向长短期记忆（Bidirectional Long Short-Term Memory，BiLSTM）[80, 92] 网络被用于捕捉句内的上下文信息。对于语境句中的第 i 个单词 $x_i (i = 1, \ldots, n)$ 来说，其

对应的隐藏状态 $\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i]$ 中的前向和后向表示分别由以下公式获得：

$$\overrightarrow{\mathbf{h}}_i = \overrightarrow{\mathbf{LSTM}}(\phi(x_i); \overrightarrow{\mathbf{h}}_{i-1}), \tag{5.6}$$

$$\overleftarrow{\mathbf{h}}_i = \overleftarrow{\mathbf{LSTM}}(\phi(x_i); \overleftarrow{\mathbf{h}}_{i+1}). \tag{5.7}$$

当前句中的第 j 个单词 $y_j (j = 1, \ldots, m)$ 对应的隐藏状态 $\mathbf{u}_j$ 也可以通过类似方法获得。值得注意的是，$\mathbf{x}$ 和 $\mathbf{y}$ 共享同一个 BiLSTM 模型。

BiLSTM 只能捕捉句内的上下文信息，但无法建模句间交互。因此，注意力层被引入以捕获语境句和当前句的关系。设 $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$ 和 $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m]$ 分别表示两句话对应的隐藏状态矩阵，相似性矩阵（Similarity Matrix）$\mathbf{A}$ 可以通过两个矩阵的如下交互得到：

$$\mathbf{A} = \cos(\mathbf{H}^\top \mathbf{U}), \tag{5.8}$$

其中 $\mathbf{A} \in \mathbb{R}^{n \times m}$，而矩阵的第 i 行，第 j 列的值 $\mathbf{A}_{i,j}$ 代表着词 x_i 和 y_j 之间的相似度。在 $\mathbf{A}$ 上按行和按列各使用一次 Softmax 函数，SplitNet 即可得到语境对当前的注意力和当前对语境的注意力。语境对当前的注意力使用 y_j 和 $\mathbf{x}$ 中每个词的相似度来补充 y_j 的表征。具体而言，令 $\mathbf{f}_j = \text{Softmax}(\mathbf{A}_{:,j})$，其中 $\mathbf{f}_j \in \mathbb{R}^n$ 表示 $\mathbf{x}$ 关于 y_j 的注意力权重。接着 y_j 就可以被一个语境可感知的向量 $\tilde{\mathbf{u}}_j = \sum_{k=1}^{n} \mathbf{f}_j[k] \cdot \mathbf{h}_k$ 来表示。当前对语境的注意力也通过类似的方式计算 $\mathbf{y}$ 关于 x_i 的注意力权重，然后将 x_i 表示为 $\tilde{\mathbf{h}}_i$。

注意力层后，SplitNet 以如下方式得到每个单词 x_i 和 y_j 的隐藏表示：

$$\mathbf{c}^{x_i} = [\mathbf{h}_i;\ \mathbf{h}_i \odot \tilde{\mathbf{h}}_i;\ \mathbf{h}_{i+1} \odot \tilde{\mathbf{h}}_{i+1}], \tag{5.9}$$

$$\mathbf{c}^{y_j} = [\mathbf{u}_j;\ \mathbf{u}_j \odot \tilde{\mathbf{u}}_j;\ \mathbf{u}_{j+1} \odot \tilde{\mathbf{u}}_{j+1}], \tag{5.10}$$

其中 $i \in \{1, \ldots, n-1\}$，$j \in \{1, \ldots, m-1\}$，$\odot$ 表示逐点乘积。令 $\mathbf{c} = (\mathbf{c}_t)_{t=1}^{T}$ 表示最终得到的隐藏状态序列，该序列可以展开成 $(\mathbf{c}^{x_1}, \ldots, \mathbf{c}^{x_{n-1}}, \mathbf{c}^{y_1}, \ldots, \mathbf{c}^{y_{m-1}})$。其中，第 t 个位置处被拆分的概率是 $\sigma(\mathbf{W} * \mathbf{c}_t + b)$，其中 σ 表示 Sigmoid 函数，$\{\mathbf{W}, b\}$ 是模型的参数。

(2) 拆分模型训练

众所周知，使用强化学习算法从随机初始化开始优化模型比较困难。因此，STAR 首先通过启发式算法从已有的数据导出带噪的训练数据以初始化 SplitNet，然后再使用强化学习算法优化它。通过识别 $(\mathbf{x}, \mathbf{y})$ 和 $\mathbf{z}$ 之间的公共子串，模型可以得到用于训练的带噪标注 $\mathbf{a}$。每个 $\mathbf{a}$ 都是一个决策序列，其中每个决策或者是拆分，或者是保留，而

$(\mathbf{x}, \mathbf{y}, \mathbf{a})$ 就构成了一个训练样本。令 $\mathcal{D}_{\text{pre}}$ 为这些训练样本组合得到的带噪训练集。在带噪训练阶段，令 θ 表示模型的参数，模型的目标函数 $\mathcal{L}_{\text{pre}}(\theta)$ 如下：

$$\mathcal{L}_{\text{pre}}(\theta) = \mathbb{E}_{(\mathbf{x},\mathbf{y},\mathbf{a})\sim\mathcal{D}_{\text{pre}}}[\log p_\theta(\mathbf{a}|\mathbf{x}, \mathbf{y})], \tag{5.11}$$

在带噪训练之后，STAR 把决策序列视作一个隐变量 $\tilde{\mathbf{a}}$，通过采样加奖励来优化。具体地，通过策略梯度方法[95]，奖励函数 $R(\tilde{\mathbf{a}}, \mathbf{z})$ (将在5.4.3节中展开说明) 被用来优化 θ 参数。此时，模型的目标函数 $\mathcal{L}_{\text{rl}}(\theta)$ 如下：

$$\mathcal{L}_{\text{rl}}(\theta) = \mathbb{E}_{(\mathbf{x},\mathbf{y},\mathbf{z})\sim\mathcal{D}}[\mathbb{E}_{\tilde{\mathbf{a}}\sim p_\theta(\tilde{\mathbf{a}}|\mathbf{x},\mathbf{y})} R(\tilde{\mathbf{a}}, \mathbf{z})]. \tag{5.12}$$

在实践中，由于候选空间遍历的计算成本过高，STAR 使用 REINFORCE 算法[96]，通过从概率分布 $p_\theta(\tilde{\mathbf{a}}|\mathbf{x}, \mathbf{y})$ 采样 M 次 $\tilde{\mathbf{a}}$ 来近似计算公式 (5.12)，其中 M 是超参数采样次数。为了优化稳定性，算法中还应用了减去基线奖励的技巧[97]。最终模型的目标函数 $\mathcal{L}_{\text{rl}}(\theta)$ 如下：

$$\mathcal{L}_{\text{rl}}(\theta) = \mathbb{E}_{(\mathbf{x},\mathbf{y},\mathbf{z})\sim\mathcal{D}}\Big[\sum_{i=1}^{M} p_\theta(\tilde{\mathbf{a}}_i|\mathbf{x}, \mathbf{y})(R(\tilde{\mathbf{a}}_i, \mathbf{z}) - \bar{R})\Big], \tag{5.13}$$

其中 $\bar{R} = \frac{1}{M}\sum_{i=1}^{M} R(\tilde{\mathbf{a}}_i, \mathbf{z})$。

5.4.3 重组阶段

本节通过两个研究问题具体地介绍了 STAR 模型的第二阶段：(1) 当得到采样的决策序列 $\tilde{\mathbf{a}}$ 后，如何计算奖励函数 $R(\tilde{\mathbf{a}}, \mathbf{z})$; (2) 如何对 P_{rec} 进行训练和推理。

(1) 奖励计算

收到决策序列 $\tilde{\mathbf{a}}$，STAR 首先枚举出所有语义冲突的候选对。以图 47为例，一旦得到一个确定的决策序列 $\tilde{\mathbf{a}}$，$(\mathbf{x}, \mathbf{y})$ 上的拆分方式就可以确定。图中 $\mathbf{x}$ 和 $\mathbf{y}$ 分别被拆分成 3 个和 2 个片段。将片段视为单位，STAR 可以系统性地枚举所有语义冲突的候选方式。枚举过程遵守一个原则，即片段之间的冲突是一对一的，这意味着一个片段或者与其他片段没有任何语义冲突（图 47中表示为 EMPTY），或者只会与另一个语句中的一个片段产生语义冲突。令 C 表示所有冲突候选方式的集合，图 47中该集合的大小是 13。

对于每个语义冲突，STAR 通过**重写过程**确定性地生成一个重写句。一般地，用当前句中与语境句中有语义冲突的片段替换原片段，STAR 以语境句为基础得到重写句。

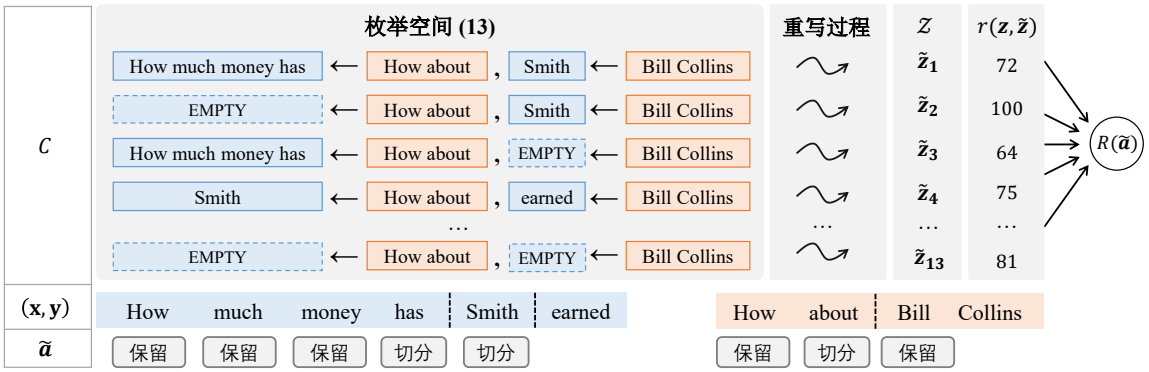

图 47　第二阶段奖励计算示意图

比如，在图 47中，C 中的第一行对应着重写句 “How about Bill Collins earned”。对于当前句中的其他片段，如果它们包含了一些关键信息如表格的列名或表格单元格的值，它们会被附加到语境句的尾部。这样做的目的是为了补救没有冲突的情况（例如，嵌套查询场景中 “Which opponent received over 537 attendance” 作为语境句，而 “And which got the result won 5-4” 作为当前句）。然而，如果当前句的某个片段包含代词，StAR 将以当前句为基础得到重写句。

在得到模型预测的重写句后，奖励函数就可以计算。StAR 使用5.3.2节中介绍的 BLEU 和 SymAcc 来构建奖励函数。最终，公式 (5.5)中的 $r(\mathbf{z},\tilde{\mathbf{z}})$ 可以通过 BLEU 和 SymAcc 的线性加和计算得出，它们的权重分别用超参数 α 和 β 表示。

(2) 重组模型训练

除了奖励的计算外，重组模型 P_{rec} 还要被训练以最大化公式 (5.5)中的目标函数。为了实现它，本节定义了一个冲突概率矩阵 $\mathbf{F}\in\mathbb{R}^{N_x\times N_y}$，其中 N_x 与 N_y 分别代表 $\mathbf{x}$ 与 $\mathbf{y}$ 中片段的数量。矩阵中第 u 行 v 列的值 $\mathbf{F}_{u,v}$ 代表了 $\mathbf{x}$ 的第 u 个片段和 $\mathbf{y}$ 的第 v 个片段冲突的概率，而该概率是在两个片段的表示上算余弦相似度（Cosine Similarity）得到的。借鉴前人工作[158]，StAR 使用 BiLSTM 两个位置上隐藏状态相减得到的向量，表示分别以这两个位置为头和尾的片段表征。例如，对于片段 $(x_i,\ldots,x_k)$ 来说，它可以用 $[\overrightarrow{\mathbf{h}_k}-\overrightarrow{\mathbf{h}_i};\overleftarrow{\mathbf{h}_i}-\overleftarrow{\mathbf{h}_k}]$ 来表示。给定一种语义冲突的方式如 $\tilde{c}\in C$，产生对应的 $\tilde{\mathbf{z}}$ 的概率 $P_{\text{rec}}(\tilde{\mathbf{z}}|\tilde{\mathbf{a}})$ 就可以写成 $g(u,v)$ 上的概率乘积：

$$P_{\text{rec}}(\tilde{\mathbf{z}}|\tilde{\mathbf{a}})=P(\tilde{c}|\mathbf{F})=\prod_{u=1}^{N_x}\prod_{v=1}^{N_y}g(u,v). \tag{5.14}$$

值得说明的是，如果 $\mathbf{x}$ 中第 u 个片段和 $\mathbf{y}$ 中第 v 个片段存在语义冲突，则令 $g(u,v)=\mathbf{F}_{u,v}$，否则的话 $g(u,v)=1-\mathbf{F}_{u,v}$。有了以上公式，就可以通过自动微分端到端地优化公式 (5.5)。为了减小计算代价，训练阶段模型只需最大化公式 (5.5)的次优解 $P_{\text{rec}}(\tilde{\mathbf{z}}^*|\tilde{\mathbf{a}})$ 即可，其中

$\tilde{\mathbf{z}}^* = \arg\max_{\tilde{\mathbf{z}} \in \mathcal{Z}}(r(\mathbf{z}, \tilde{\mathbf{z}}))$，表示模型所预测的最接近标注的重写句。

(3) 重组模型推理

在训练阶段，模型可以在标注重写句 $\mathbf{z}$ 的指导下（如计算预测重写句和标注重写句之间的相似度）找到合适的 $\tilde{\mathbf{z}}^*$。然而在推理阶段，因为并没有标注的重写句，模型只能从 $\mathbf{F}$ 中得到 $\tilde{\mathbf{z}}^*$。具体地，对于当前句中第 v 个片段，模型先找到语境句中与其最可能冲突的片段 $u^* = \arg\max_u \mathbf{F}_{u,v}$。接着，模型引入超参数 λ 作为阈值以判定语义冲突是否存在。当 $\mathbf{F}_{u^*,v} >= \lambda$ 时，模型判定当前句中第 v 个片段与 u^* 存在语义冲突，否则判定为无语义冲突。最终，按照如上所述的重写过程，重组模型可以通过语义冲突的结果确定性地得到一个重写句。

5.5 实验与验证

本节通过 FollowUp 数据集上充分的实验来验证 STAR 方法的有效性，其中数据集与评价指标等细节请参考5.3节。

5.5.1 实验设置

五个强大的序列模型被选作 STAR 的基线模型。其中包括了**基于规则**的、**基于序列生成**的以及**基于指代消解**的三大种类：

CONCAT：最简单的基于规则的基线模型，它直接将语境句和当前句拼接在一起作为重写句，这种做法可以最大程度地保留信息，但生成的重写句与语境无关句子的风格相差较远。

SEQ2SEQ：带注意力的序列到序列模型[27]，是机器翻译中最常用的基线模型之一。

COPYNET：带复制机制的序列到序列模型[82]，可以从语境句和当前句中学会复制信息用在重写句中。

COPYNET+BERT：在 COPYNET 的基础上使用最先进的预训练模型 BERT[41] 初始化其编码器。

E2ECR：端到端进行指代消解的模型[159]。值得注意的是，实验时本节直接使用原作者提供的训练好的模型权重在 FollowUp 数据集上进行评估，因为 FollowUp 数据集没有细粒度的指代关系用于该模型的训练。

PyTorch[105] 和 AllenNLP[160] 被使用来实现 STAR，模型的词嵌入和隐藏状态的维度都是 100。模型的优化使用 Adam[161] 作为优化器，带噪数据训练时模型的学习率被设定为 1×10^{-3}，强化学习中模型的学习率则调低为 1×10^{-4}。在 REINFORCE 算法的实现

中，超参数采样数量 M 设置为 20。对于其他超参数，α 设置为 0.5，β 设置为 0.5，以及 λ 设置为 0.6。

5.5.2　实验结果

表 26　FollowUp 数据集上不同方法的性能

模型	开发集		测试集		
	SymAcc （%）	BLEU （%）	SymAcc （%）	BLEU （%）	ExAcc （%）
Seq2Seq[27]	0.63 ± 0.00	21.34 ± 1.14	0.50 ± 0.22	20.72 ± 1.31	–
CopyNet[82]	17.50 ± 0.87	43.36 ± 0.54	19.30 ± 0.93	43.34 ± 0.45	–
CopyNet+BERT[41]	18.63 ± 0.61	45.14 ± 0.68	22.00 ± 0.45	44.87 ± 0.52	–
Concat	–	–	22.00 ± –	52.02 ± –	25.24
E2ECR[159]	–	–	27.00 ± –	52.47 ± –	27.18
StAR（本方法）	**55.38** ± 1.21	**67.62** ± 0.65	**54.00** ± 1.09	**67.05** ± 1.05	**65.05**

表 26显示了不同模型在 FollowUp 数据集开发集和测试集上的 SymAcc、BLEU 和 ExAcc，表格中呈现的是基线模型与本方法在 5 个随机种子下实验结果的均值和标准差。从 SymAcc 和 BLEU 来看，正如表格中所呈现的，StAR 在这两个指标上明显优于所有基线。例如，在测试集上，StAR 相比最先进的基线 E2ECR 在 BLEU 上获得了高达 14.58% 的绝对改进。表格中的结果同时表明，在原句上进行改写的方式对于对话重写任务来说更加合理，因为即使是最简单的基于规则的 Concat，也比基于序列生成的的基线模型如 CopyNet 表现更好。

从 ExAcc 来看，与极有竞争力的基线模型相比，StAR 同样达到了最好的性能 65.05%，这表明了它相比基线模型的优越性。而与最简单的基线模型 Concat 相比，本方法额外释放了单轮语义解析模型 Coarse2Fine 在对话式语义解析上超过 39.81% 的性能。这一结果有力地验证了对话重写任务的必要性，以及它和单轮语义解析器合作完成半监督下对话式语义解析任务的可行性。

5.5.3　实验分析

(1) 变体分析

本节探索了 StAR 的不同变体以验证模型各个部分设计的合理性。如表 27所示，有三个部分在变体分析中被分别消融：

– **第一阶段** 不再切分，即意味着模型在词粒度而非片段粒度执行第二阶段。

表 27　FollowUp 开发集上 StAR 模型不同变体的实验结果

模型变体	SymAcc （%）	BLEU （%）
原始	55.38	67.62
– 第一阶段	40.63	61.82
– 第二阶段	23.12	48.65
– 强化学习	41.25	60.19
+ 标准奖励	43.13	58.48
+ 真值奖励	45.20	63.04
+ 均匀奖励	53.40	66.93

– 第二阶段 在测试时，在重组阶段只进行随机猜测，而不用模型预测的结果。

– 强化学习 仅有带噪训练阶段，而没有额外的强化学习训练过程。

从表格中可以看出，当消融掉第一阶段时，关键词保留率 SymAcc 从 55% 跌到了 40%。消融第二阶段造成的下降更严重，模型仅能获得 23% 的关键词保留率。这些显著下降的性能表明第一阶段和第二阶段对模型都非常重要。去掉强化学习阶段后模型的性能也很差，验证了使用强化学习训练 StAR 的必要性。

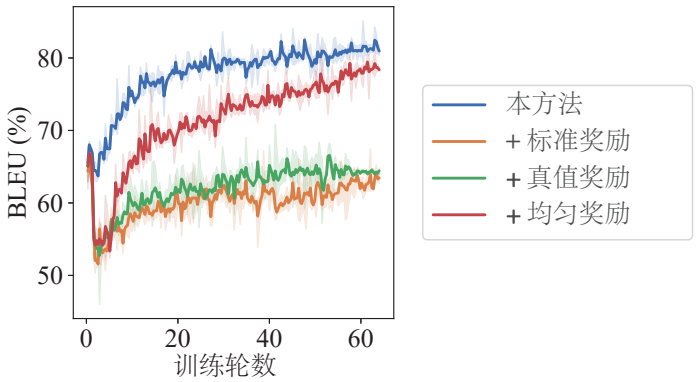

图 48　FollowUp 数据集上不同变体在训练集上的学习曲线，StAR 的收敛速度和收敛稳定性比其他变体更好

为了验证公式 (5.5)作为奖励函数 $R(q, \mathbf{z})$ 的合理性，本节实验了关于 $R(q, \mathbf{z})$ 的三个变体奖励函数：

+ 标准奖励 原始 REINFORCE 策略，即同时采样 $q \in \mathcal{Q}$ 和 $\tilde{\mathbf{z}} \in \mathcal{Z}$，然后将 $r(\mathbf{z}, \tilde{\mathbf{z}})$ 作为 $R(q, \mathbf{z})$。

+ 真值奖励 假定了语义冲突过程总是正确，使用 $\max_{\tilde{\mathbf{z}} \in \mathcal{Z}}(r(\mathbf{z}, \tilde{\mathbf{z}}))$ 计算 $R(q, \mathbf{z})$。

+ 均匀奖励 给所有的 $\tilde{\mathbf{z}}$ 赋予了相同的采样概率，使用它们的均值 $\sum_{\tilde{\mathbf{z}} \in \mathcal{Z}} r(\mathbf{z}, \tilde{\mathbf{z}}) \, |\mathcal{Z}|$ 来

计算 $R(q, \mathbf{z})$。

实验结果如表 27所示，可以看出 STAR 所采用的奖励函数取得了最好的效果。正如4.2节中所提到的，直接使用原始 REINFORCE 策略动作空间太大，会导致标准奖励最终的效果不好。通过牺牲一定计算量枚举 $\tilde{\mathbf{z}}$，STAR 将动作空间从 $|\mathcal{Q}| \cdot |\mathcal{Z}|$ 降低到了 $|\mathcal{Q}|$ 大小。如图 48所示，STAR 的收敛速度和收敛稳定性也都比其他奖励变体更好。量化的模型收敛速度测试表明，STAR 所采用的奖励函数可以比标准奖励的收敛时间快将近 15 倍，采样效率非常高。

(2) 样例分析

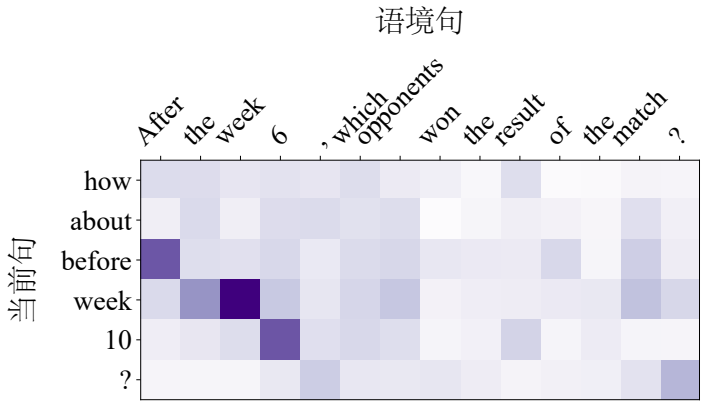

图 49　在真实样例上拆分模型的相似度矩阵可视化结果，颜色越深表明相似度越高

图 49显示了拆分模型在真实样例，即语境句 “After the week 6, which opponents won the result of the match?”（**翻译：第六周后，哪些选手赢得了比赛结果？**）和当前句 “how about before week 10? ”（**翻译：那第 10 周前呢？**）上产生的相似性矩阵 **A**。从图中可以明显看出，片段 “Afther the week 6” 与片段 “before week 10” 之间的相似度明显更高，说明拆分模型捕捉到了这两个片段间的语义冲突，符合对模型的预期。

图 50中展示了三个真实样例上 STAR 预测的重写句。语境句中每个片段的颜色深浅代表其与当前句中标记为蓝色的片段语义冲突的概率大小，颜色越深代表概率越高。在第一个样例中，“Glebe park”，“Hampden park” 和 “Balmoor” 都是数据库表中的单元格值，并处于同一列。STAR 以很高的概率正确发现了语义冲突对 “compared to Glebe park” 和 “compared to Balmoor”。第二个样例展示了 STAR 的灵活性，其中 “the writer Nancy miller” 需要作为一个整体片段被拆分出来，以整体替换掉语境句中的 “Nancy miller”。至于第三个样例，STAR 识别出代词 “those two films” 指代的是 “Greatest Love and Promised Land”，成功地完成了指代消解功能。由上述样例可知，受益于两个阶段的设计，STAR

样例 1

语境句 : [compared to Glebe park] [, does] [Hampden park] [holds more attendances at capacity ?]
翻译 : 与Glebe公园相比，Hampden公园是不是能容纳更多的游客?
当前句 : [how about] [compared to Balmoor]
翻译 : 与Balmoor相比如何?
预测重写句 : compared to Balmoor , does Hampden park holds more attendances at capacity ?
翻译 : 与Balmoor公园相比，Hampden公园是不是能容纳更多的游客?

样例 2

语境句 : [Is there any book which belongs to] [Nancy miller]
翻译 : 有没有属于南希米勒的书?
当前句 : [I mean] [the writer Nancy miller]
翻译 : 我是说作家南希米勒。
预测重写句 : Is there any book which belongs to the writer Nancy miller
翻译 : 有没有属于作家南希米勒的书?

样例 3

语境句 : [show directors of] [Greatest Love and Promised Land]
翻译 : 《最佳爱情》和《应许之地》的导演是谁?
当前句 : [show air date of] [those two films]
翻译 : 这两部电影是什么时候发行的?
预测重写句 : show air date of greatest love and promised land
翻译 : 《最佳爱情》和《应许之地》是什么时候发行的?

图 50　FollowUp 数据集上 STAR 预测结果的样例分析，语境句中每个片段的颜色深浅代表其与当前句中标记为蓝色的片段语义冲突的概率大小，颜色越深代表冲突概率越高

能够灵活地处理各种对话重写的情形。

(3) 错误分析

STAR 在大多数情况下都运行良好，但也有一些情况模型在切分阶段（即第一阶段）会出现失误。例如，给定一个语境句如 “面积最大的区域在哪？” 和当前句 “那最小的呢”，STAR 更倾向于将 “最大的区域” 识别为一个片段，而不是所预期的将 “最大的” 和 “区域” 识别为两个片段。这可能是因为 “最大的区域” 这个短语出现频次较高，STAR 较难识别出在该场景下 “最大的” 的语义要从原句中拆分出来。

5.6　本章小结

在本章中，为提升对话式语义解析模型在半监督下的性能，本章提出将对话式语义解析解耦为对话重写和单轮语义解析两个子任务，从而利用已有的单轮语义解析数据资源，搭配标注成本低廉的对话重写数据来解决对话式语义解析场景。为实现该目标，本章构建了首个面向语义解析的对话重写数据集 FollowUp，该数据集由跨 120 张表格的 1,000 个对话组成。为更好地完成对话重写任务，本章提出一个基于拆分重组的对话重写方法，通过直接编辑对话更好地利用对话本身的信息。通过引入片段作为对话的基本单位，该方法将对话重写设计为一个两阶段的过程，在将原始对话拆分成各个片段后，

再通过片段的重组完成对话重写任务。在 FollowUp 数据集上的实验表明，对话重写任务用于驱动半监督下对话式自然语言语义解析是可行的，且基于拆分重组的方法可以比前人基线模型更好地发挥语义解析模型的性能。最终，本章所提出的方法可以在让对话式语义解析模型在半监督下取得全监督训练模型 65% 的效果。

局限性与未来工作　虽然本方法利用对话重写推动了对话式语义解析模型在半监督下的效果，但是由于对话重写生成的自然语言与已有的单轮语义解析训练时见过的自然语言分布不同，导致一些情况下即使对话重写生成了正确的重写句，单轮语义解析模型却无法解析成正确的程序。未来的工作是在不引入额外数据的情况下联合训练对话重写模型与单轮语义解析模型，从而使两个模块可以更好地协作搭配。

总结与展望

工作总结

自然语言语义解析是自然语言处理与人工智能领域的一个前沿课题。通过将用户输入的自然语言解析成形式化的计算机程序，语义解析可以支持复杂的下游任务，让用户仅通过自然语言就可以指挥机器完成任务。然而，语义解析中程序标注需要耗费大量的人力和财力，阻碍了该领域的规模化发展，弱标注是解决该问题的重要方向之一。因此，本文立足于自然语言语义解析，针对弱标注下的自然语言语义解析所面临的诸多难点开展了一系列自然语言语义解析的研究。具体而言，本文的研究工作与成果总结如下：

1、为提升语义解析模型面向程序的组合泛化能力，受启发于人类的层次化抽象思维，本文提出了一个记忆单元增强的神经网络架构，同时提出分层强化学习算法和课程学习策略成功训练该架构。在组合泛化著名的评测基准上进行的实验表明，该架构以100% 的准确率解决了前人所提出的组合泛化挑战。本方法也成为首个不需要任何人工规则就可以解决前人所提出的组合泛化挑战的基于神经网络的方法。

2、为提升语义解析模型面向知识库领域的泛化能力，本文提出了一种通过擦除，仅使用语义解析数据即可训练实体链接模型的方法。在四个实体链接数据集上的实验结果表明，该方法所训练出的实体链接模型效果甚至可以匹配全监督训练出的模型。在两个经典的跨域语义解析数据集上的实验结果表明，本方法可以灵活地应用到现有的语义解析器中，并显著提高它们的领域泛化能力。

3、为提升弱监督下答案驱动的自然语言语义解析方法的性能，本文提出了一种使用生成式模型可微地解决弱监督语义解析的方法，该生成式模型可以被视为包含了一个潜在的语义解析模型和一个潜在的程序执行器。为进一步提升该模型的性能，本文提出了一种执行引导的预训练方法。三个弱监督语义解析数据集上的实验结果表明，使用生成式模型解决弱监督语义解析任务非常有效，且本文所提出的预训练方法可以极大提升弱监督语义解析场景下生成式模型的性能，最终性能甚至与强监督下最好的基线模型性能持平。

4、为降低对话式语义解析模型的构建难度，本文提出将对话式语义解析任务解耦为对话重写和单轮语义解析。通过众包标注涵盖多个不同场景的对话重写数据集，本文可以复用已有的单轮语义解析数据，实现半监督下的对话式语义解析。面向对话重写，本文提出一种可编辑对话的基于拆分重组的方法。在本文标注的对话重写数据集上的实验表明，对话重写驱动半监督地解决对话式语义解析模型是可行的，而且本方法可以更

好地释放模型在对话式语义解析场景的性能。

工作展望

本文立足于自然语言语义解析，针对弱标注下的自然语言语义解析所面临的难点，从程序组合泛化性提升，知识库领域泛化性提升，弱监督下答案驱动语义解析模型性能提升，半监督下快速构建对话式语义解析模型等方面开展了一系列研究。然而，自然语言语义解析仍有大量未解决的挑战亟待研究人员探索，本文在此对未来研究工作提出几点展望：

1、异质知识库下的自然语言语义解析模型。虽然现在语义解析模型在各种不同知识库上已经有了成功应用，但目前的语义解析模型仍然只能一次性使用一种知识库，这阻碍了语义解析模型处理更复杂任务能力的发展。因此，在需要多种不同知识库的场景，即异质知识库下的自然语言语义解析系统是未来很有前景的一个研究方向。与只能使用特定知识库进行建模的现有系统相比，异质知识库下的自然语言语义解析模型提供了更多的可能性。它既可以纳入来自结构化知识库的领域相关知识，也可以容纳来自非结构化知识源的世界知识，从而生成表达能力更强的程序，完成更加复杂的下游任务。

2、将自然语言语义解析方法扩展到非结构化文档。目前主流的语义解析模型主要关注在结构化的知识库场景，比如知识图谱或者数据库。然而，现实生活中用户接触更多的是非结构化的文档，在这些文档上构建下游应用可以更好地服务用户，产生商业价值。因此，通过开发文档适用的程序语言和程序执行器将自然语言语义解析方法扩展到非结构化文档是另一个非常有前景的研究方向。

3、可交互的和可解释的自然语言语义解析模型。对话式语义解析中的核心问题除了语境理解外，还应该支持人机交互的另外一种形式，即用户作为老师，语义解析模型可以通过用户的反馈修正自身错误。可解释性是目前人工智能领域最热点的话题，一个好的语义解析模型应该不仅能够支持用户解决复杂任务，还应该能够产生用户可以看懂的解释以获得用户的信赖。然而，当前绝大部分工作都忽略了语义解析模型的可交互性和可解释性。因此，如何推进语义解析的可交互性和可解释性也是重要的研究方向。

参 考 文 献

[1] GUPTA S, SHAH R, MOHIT M, et al. Semantic parsing for task oriented dialog using hierarchical representations[C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 2787-2792. https://aclanthology.org/D18-1300. DOI: 10.18653/v1/D18-1300.

[2] GUO J, ZHAN Z, GAO Y, et al. Towards complex text-to-SQL in cross-domain database with intermediate representation[C/OL]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 4524-4535. https://aclanthology.org/P19-1444. DOI: 10.18653/v1/P19-1444.

[3] LIN X V, WANG C, ZETTLEMOYER L, et al. Nl2bash: A corpus and semantic parser for natural language interface to the linux operating system[C/OL]//Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018. European Language Resources Association (ELRA), 2018. http://www.lrec-conf.org/proceedings/lrec2018/summaries/1021.html.

[4] MATUSZEK C, HERBST E, ZETTLEMOYER L S, et al. Learning to parse natural language commands to a robot control system[C/OL]//Proceedings of the 13th International Symposium on Experimental Robotics, ISER 2012, June 18-21, 2012, Québec City, Canada. Springer, 2012: 403-415. https://doi.org/10.1007/978-3-319-00065-7_28.

[5] ZELLE J M, MOONEY R J. Learning to parse database queries using inductive logic programming[C/OL]//Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, IAAI 96, Portland, Oregon, USA, August 4-8, 1996, Volume 2. AAAI Press / The MIT Press, 1996: 1050-1055. http://www.aaai.org/Library/AAAI/1996/aaai96-156.php.

[6] TANG L R, MOONEY R J. Using multiple clause constructors in inductive logic programming for semantic parsing[C/OL]//Machine Learning: EMCL 2001, 12th European Conference on Machine Learning, Freiburg, Germany, September 5-7, 2001, Proceedings. Springer, 2001: 466-477. https://doi.org/10.1007/3-540-44795-4_40.

[7] HEMPHILL C T, GODFREY J J, DODDINGTON G R. The ATIS spoken language systems pilot corpus[C/OL]//Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, USA, June 24-27, 1990. Morgan Kaufmann, 1990. https://aclanthology.org/H90-1021/.

[8] WANG Y, BERANT J, LIANG P. Building a Semantic Parser Overnight[C/OL]// 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). Beijing, China: Association for Computational Linguistics, 2015: 1332-1342. https://aclanthology.org/P15-1129. DOI: 10.3115/v1/P15-1129.

[9] CAI Q, YATES A. Semantic parsing Freebase: Towards open-domain semantic parsing[C/OL]//Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity. Atlanta, Georgia, USA: Association for Computational Linguistics, 2013: 328-338. https://aclanthology.org/S13-1045.

[10] YU T, ZHANG R, YANG K, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task[C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 3911-3921. https://aclanthology.org/D18-1425. DOI: 10.18653/v1/D18-1425.

[11] ZHONG V, XIONG C, SOCHER R. Seq2sql: Generating structured queries from natural language using reinforcement learning[J]. CoRR, 2017, abs/1709.00103.

[12] LING W, BLUNSOM P, GREFENSTETTE E, et al. Latent predictor networks for code generation[C/OL]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, 2016: 599-609. https://aclanthology.org/P16-1057. DOI: 10.18653/v1/P16-1057.

[13] ODA Y, FUDABA H, NEUBIG G, et al. Learning to generate pseudo-code from source code using statistical machine translation (T)[C/OL]//30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015, Lincoln, NE, USA, November 9-13, 2015. IEEE Computer Society, 2015: 574-584. https://doi.org/10.1109/ASE.2015.36.

[14] YIN P, DENG B, CHEN E, et al. Learning to mine aligned code and natural language

pairs from stack overflow[C/OL]//Proceedings of the 15th International Conference on Mining Software Repositories, MSR 2018, Gothenburg, Sweden, May 28-29, 2018. ACM, 2018: 476-486. https://doi.org/10.1145/3196398.3196408.

[15] CHEN M, TWOREK J, JUN H, et al. Evaluating large language models trained on code [J/OL]. CoRR, 2021, abs/2107.03374. https://arxiv.org/abs/2107.03374.

[16] LI Y, CHOI D H, CHUNG J, et al. Competition-level code generation with alphacode [J/OL]. CoRR, 2022, abs/2203.07814. https://doi.org/10.48550/arXiv.2203.07814.

[17] BOBROW D G. Natural language input for a computer problem solving system[J]. 1964.

[18] ZETTLEMOYER L S, COLLINS M. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars[C/OL]//UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005. AUAI Press, 2005: 658-666. https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1209&proceeding_id=21.

[19] ZETTLEMOYER L, COLLINS M. Online learning of relaxed CCG grammars for parsing to logical form[C/OL]//Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). Prague, Czech Republic: Association for Computational Linguistics, 2007: 678-687. https://aclanthology.org/D07-1071.

[20] 陈波, 孙乐, 韩先培. 基于桥连接的词典学习方法的语义解析[J/OL]. 中文信息学报, 2019, 33(5):24. http://jcip.cipsc.org.cn/CN/abstract/article_2761.shtml.

[21] LIANG P, JORDAN M I, KLEIN D. Learning dependency-based compositional semantics[J/OL]. Computational Linguistics, 2013, 39(2):389-446. https://aclanthology.org/J13-2005. DOI: 10.1162/COLI_a_00127.

[22] CHEN D, MANNING C. A fast and accurate dependency parser using neural networks [C/OL]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014: 740-750. https://aclanthology.org/D14-1082. DOI: 10.3115/v1/D14-1082.

[23] LIANG P. Lambda dependency-based compositional semantics[J/OL]. CoRR, 2013, abs/1309.4408. http://arxiv.org/abs/1309.4408.

[24] DONG L, LAPATA M. Language to logical form with neural attention[C/OL]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Lin-

guistics, 2016: 33-43. https://aclanthology.org/P16-1004. DOI: 10.18653/v1/P16-1004.

[25] ANDREAS J, VLACHOS A, CLARK S. Semantic parsing as machine translation [C/OL]//Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Sofia, Bulgaria: Association for Computational Linguistics, 2013: 47-52. https://aclanthology.org/P13-2009.

[26] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C/OL]//GHAHRAMANI Z, WELLING M, CORTES C, et al. Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. 2014: 3104-3112. https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html.

[27] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[C/OL]//BENGIO Y, LECUN Y. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015. http://arxiv.org/abs/1409.0473.

[28] DONG L, LAPATA M. Coarse-to-fine decoding for neural semantic parsing[C/OL]// Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, 2018: 731-742. https://aclanthology.org/P18-1068. DOI: 10.18653/v1/P18-1068.

[29] YIN P, NEUBIG G. TRANX: A transition-based neural abstract syntax parser for semantic parsing and code generation[C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Brussels, Belgium: Association for Computational Linguistics, 2018: 7-12. https://aclanthology.org/D18-2002. DOI: 10.18653/v1/D18-2002.

[30] RABINOVICH M, STERN M, KLEIN D. Abstract syntax networks for code generation and semantic parsing[C/OL]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, 2017: 1139-1149. https://aclanthology.org/P17-1105. DOI: 10.18653/v1/P17-1105.

[31] KRISHNAMURTHY J, DASIGI P, GARDNER M. Neural semantic parsing with type constraints for semi-structured tables[C/OL]//Proceedings of the 2017 Conference on

Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, 2017: 1516-1526. https://aclanthology.org/D17-1160. DOI: 10.18653/v1/D17-1160.

[32] YIN P, NEUBIG G. A syntactic neural model for general-purpose code generation [C/OL]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, 2017: 440-450. https://aclanthology.org/P17-1041. DOI: 10.18653/v1/P17-1041.

[33] SUN Y, TANG D, DUAN N, et al. Semantic parsing with syntax- and table-aware SQL generation[C/OL]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, 2018: 361-372. https://aclanthology.org/P18-1034. DOI: 10.18653/v1/P18-1034.

[34] CHEN B, SUN L, HAN X. Sequence-to-action: End-to-end semantic graph generation for semantic parsing[C/OL]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, 2018: 766-777. https://aclanthology.org/P18-1071. DOI: 10.18653/v1/P18-1071.

[35] GUO D, TANG D, DUAN N, et al. Dialog-to-action: Conversational question answering over a large-scale knowledge base[C/OL]//Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada. 2018: 2946-2955. https://proceedings.neurips.cc/paper/2018/hash/d63fbf8c3173730f82b150c5ef38b8ff-Abstract.html.

[36] YU T, YASUNAGA M, YANG K, et al. SyntaxSQLNet: Syntax tree networks for complex and cross-domain text-to-SQL task[C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 1653-1663. https://aclanthology.org/D18-1193. DOI: 10.18653/v1/D18-1193.

[37] LIN K, BOGIN B, NEUMANN M, et al. Grammar-based neural text-to-sql generation [J/OL]. CoRR, 2019, abs/1905.13326. http://arxiv.org/abs/1905.13326.

[38] WANG B, SHIN R, LIU X, et al. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers[C/OL]//Proceedings of the 58th Annual Meeting of the

Association for Computational Linguistics. Online: Association for Computational Linguistics, 2020: 7567-7578. https://aclanthology.org/2020.acl-main.677. DOI: 10.18653/v1/2020.acl-main.677.

[39] LI Y, TARLOW D, BROCKSCHMIDT M, et al. Gated graph sequence neural networks[C/OL]//4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. 2016. http://arxiv.org/abs/1511.05493.

[40] BOGIN B, GARDNER M, BERANT J. Global reasoning over database structures for text-to-SQL parsing[C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 3659-3664. https://aclanthology.org/D19-1378. DOI: 10.18653/v1/D19-1378.

[41] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[C/OL]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 4171-4186. https://aclanthology.org/N19-1423. DOI: 10.18653/v1/N19-1423.

[42] BROWN T B, MANN B, RYDER N, et al. Language models are few-shot learners [C/OL]//Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. 2020. https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

[43] RAFFEL C, SHAZEER N, ROBERTS A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J/OL]. J. Mach. Learn. Res., 2020, 21:140:1-140:67. http://jmlr.org/papers/v21/20-074.html.

[44] HERZIG J, NOWAK P K, MÜLLER T, et al. TaPas: Weakly supervised table parsing via pre-training[C/OL]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, 2020: 4320-4333. https://aclanthology.org/2020.acl-main.398. DOI: 10.18653/v1/2020.acl-main.398.

[45] YIN P, NEUBIG G, YIH W T, et al. TaBERT: Pretraining for joint understanding of textual and tabular data[C/OL]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, 2020: 8413-8426. https://aclanthology.org/2020.acl-main.745. DOI: 10.18653/v1/2020.acl-main.745.

[46] YU T, WU C, LIN X V, et al. Grappa: Grammar-augmented pre-training for table semantic parsing[C/OL]//9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. https://openreview.net/forum?id=kyaIeYj4zZ.

[47] JIA R, LIANG P. Data recombination for neural semantic parsing[C/OL]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, 2016: 12-22. https://aclanthology.org/P16-1002. DOI: 10.18653/v1/P16-1002.

[48] SHIN R, LIN C H, THOMSON S, et al. Constrained language models yield few-shot semantic parsers[C/OL]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. Association for Computational Linguistics, 2021: 7699-7715. https://doi.org/10.18653/v1/2021.emnlp-main.608.

[49] POESIA G, POLOZOV O, LE V, et al. Synchromesh: Reliable code generation from pre-trained language models[J/OL]. CoRR, 2022, abs/2201.11227. https://arxiv.org/abs/2201.11227.

[50] RUSSIN J L, JO J, O'REILLY R C, et al. Compositional generalization by factorizing alignment and translation[C/OL]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, ACL 2020, Online, July 5-10, 2020. Association for Computational Linguistics, 2020: 313-327. https://doi.org/10.18653/v1/2020.acl-srw.42.

[51] LI Y, ZHAO L, WANG J, et al. Compositional generalization for primitive substitutions [C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 4293-4302. DOI: 10.18653/v1/D19-1438.

[52] CHEN X, LIANG C, YU A W, et al. Compositional generalization via neural-symbolic

stack machines[C/OL]//Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. 2020. https://proceedings.neurips.cc/paper/2020/hash/12b1e42dc0746f22cf361267de07073f-Abstract.html.

[53] ZHENG H, LAPATA M. Compositional generalization via semantic tagging[C/OL]// Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021. Association for Computational Linguistics, 2021: 1022-1032. https://doi.org/10.18653/v1/2021.findings-emnlp.88.

[54] OREN I, HERZIG J, BERANT J. Finding needles in a haystack: Sampling structurally-diverse training sets from synthetic data for compositional generalization[C/OL]// Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. Association for Computational Linguistics, 2021: 10793-10809. https://doi.org/10.18653/v1/2021.emnlp-main.843.

[55] NYE M I, SOLAR-LEZAMA A, TENENBAUM J, et al. Learning compositional rules via neural program synthesis[C/OL]//Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. 2020. https://proceedings.neurips.cc/paper/2020/hash/7a685d9edd95508471a9d3d6fcace432-Abstract.html.

[56] ANDREAS J. Good-enough compositional data augmentation[C/OL]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, 2020: 7556-7566. https://aclanthology.org/2020.acl-main.676. DOI: 10.18653/v1/2020.acl-main.676.

[57] GORDON J, LOPEZ-PAZ D, BARONI M, et al. Permutation equivariant models for compositional generalization in language[C/OL]//8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. https://openreview.net/forum?id=SylVNerFvr.

[58] LAKE B M. Compositional generalization through meta sequence-to-sequence learning [C/OL]//Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. 2019: 9788-9798. https://proceedings.neurips.cc/paper/2019/

hash/f4d0e2e7fc057a58f7ca4a391f01940a-Abstract.html.

[59] CONKLIN H, WANG B, SMITH K, et al. Meta-learning to compositionally generalize [C/OL]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Online: Association for Computational Linguistics, 2021: 3322-3335. https://aclanthology.org/2021.acl-long.258. DOI: 10.18653/v1/2021.acl-long.258.

[60] YIN P, FANG H, NEUBIG G, et al. Compositional generalization for neural semantic parsing via span-level supervised attention[C/OL]//Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Online: Association for Computational Linguistics, 2021: 2810-2823. https://aclanthology.org/2021.naacl-main.225. DOI: 10.18653/v1/2021.naacl-main.225.

[61] BERANT J, CHOU A, FROSTIG R, et al. Semantic parsing on Freebase from question-answer pairs[C/OL]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics, 2013: 1533-1544. https://aclanthology.org/D13-1160.

[62] PASUPAT P, LIANG P. Compositional semantic parsing on semi-structured tables [C/OL]//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). Beijing, China: Association for Computational Linguistics, 2015: 1470-1480. https://aclanthology.org/P15-1142. DOI: 10.3115/v1/P15-1142.

[63] LIANG C, BERANT J, LE Q, et al. Neural symbolic machines: Learning semantic parsers on Freebase with weak supervision[C/OL]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, 2017: 23-33. https://aclanthology.org/P17-1003. DOI: 10.18653/v1/P17-1003.

[64] LIANG C, NOROUZI M, BERANT J, et al. Memory augmented policy optimization for program synthesis and semantic parsing[C/OL]//Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada. 2018: 10015-10027. https://proceedings.neurips.cc/paper/2018/hash/f4e369c0a468d3aeeda0593ba90

b5e55-Abstract.html.

[65] AGARWAL R, LIANG C, SCHUURMANS D, et al. Learning to generalize from sparse and underspecified rewards[C/OL]//Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. PMLR, 2019: 130-140. http://proceedings.mlr.press/v97/agarwal19e.html.

[66] GUO J, LOU J, LIU T, et al. Weakly supervised semantic parsing by learning from mistakes[C/OL]//Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021. Association for Computational Linguistics, 2021: 2603-2617. https://doi.org/10.18653/v1/2021.findings-emnlp.222.

[67] GUU K, PASUPAT P, LIU E, et al. From language to programs: Bridging reinforcement learning and maximum marginal likelihood[C/OL]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, 2017: 1051-1062. https://aclanthology.org/P17-1097. DOI: 10.18653/v1/P17-1097.

[68] MIN S, CHEN D, HAJISHIRZI H, et al. A discrete hard EM approach for weakly supervised question answering[C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 2851-2864. https://aclanthology.org/D19-1284. DOI: 10.18653/v1/D19-1284.

[69] NEELAKANTAN A, LE Q V, ABADI M, et al. Learning a natural language interface with neural programmer[C/OL]//5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net, 2017. https://openreview.net/forum?id=ry2YOrcge.

[70] DAHL D A, BATES M, BROWN M, et al. Expanding the scope of the ATIS task: The ATIS-3 corpus[J]. Proceedings of the workshop on Human Language Technology, 1994.

[71] YU T, ZHANG R, YASUNAGA M, et al. SParC: Cross-domain semantic parsing in context[C/OL]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 4511-4523. https://aclanthology.org/P19-1443. DOI: 10.18653/v1/P19-1443.

[72] YU T, ZHANG R, ER H, et al. CoSQL: A conversational text-to-SQL challenge towards

cross-domain natural language interfaces to databases[C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 1962-1979. https://aclanthology.org/D19-1204. DOI: 10.18653/v1/D19-1204.

[73] GUO J, SI Z, WANG Y, et al. Chase: A large-scale and pragmatic Chinese dataset for cross-database context-dependent text-to-SQL[C/OL]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Online: Association for Computational Linguistics, 2021: 2316-2331. https://aclanthology.org/2021.acl-long.180. DOI: 10.18653/v1/2021.acl-long.180.

[74] LONG R, PASUPAT P, LIANG P. Simpler context-dependent logical forms via model projections[C/OL]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, 2016: 1456-1465. https://aclanthology.org/P16-1138. DOI: 10.18653/v1/P16-1138.

[75] IYER S, KONSTAS I, CHEUNG A, et al. Mapping language to code in programmatic context[C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 1643-1652. https://aclanthology.org/D18-1192. DOI: 10.18653/v1/D18-1192.

[76] MILLER S, STALLARD D, BOBROW R, et al. A fully statistical approach to natural language interfaces[C/OL]//34th Annual Meeting of the Association for Computational Linguistics. Santa Cruz, California, USA: Association for Computational Linguistics, 1996: 55-61. https://aclanthology.org/P96-1008. DOI: 10.3115/981863.981871.

[77] ZETTLEMOYER L, COLLINS M. Learning context-dependent mappings from sentences to logical form[C/OL]//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. Suntec, Singapore: Association for Computational Linguistics, 2009: 976-984. https://aclanthology.org/P09-1110.

[78] SERBAN I V, SORDONI A, BENGIO Y, et al. Building end-to-end dialogue systems using generative hierarchical neural network models[C/OL]//Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona,

USA. AAAI Press, 2016: 3776-3784. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11957.

[79] SUHR A, IYER S, ARTZI Y. Learning to map context-dependent sentences to executable formal queries[C/OL]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics, 2018: 2238-2249. https://aclanthology.org/N18-1203. DOI: 10.18653/v1/N18-1203.

[80] SCHUSTER M, PALIWAL K K. Bidirectional recurrent neural networks[J/OL]. IEEE Trans. Signal Processing, Volume 45, 1997:2673-2681. https://doi.org/10.1109/78.650093.

[81] ZHANG R, YU T, ER H, et al. Editing-based SQL query generation for cross-domain context-dependent questions[C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 5338-5349. https://aclanthology.org/D19-1537. DOI: 10.18653/v1/D19-1537.

[82] GU J, LU Z, LI H, et al. Incorporating copying mechanism in sequence-to-sequence learning[C/OL]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, 2016: 1631-1640. https://aclanthology.org/P16-1154. DOI: 10.18653/v1/P16-1154.

[83] ANDREAS J, BUFE J, BURKETT D, et al. Task-oriented dialogue as dataflow synthesis [J/OL]. Transactions of the Association for Computational Linguistics, 2020, 8:556-571. https://aclanthology.org/2020.tacl-1.36. DOI: 10.1162/tacl_a_00333.

[84] ZAREMOODI P, HAFFARI G. Adaptively scheduled multitask learning: The case of low-resource neural machine translation[C/OL]//Proceedings of the 3rd Workshop on Neural Generation and Translation. Hong Kong: Association for Computational Linguistics, 2019: 177-186. https://aclanthology.org/D19-5618. DOI: 10.18653/v1/D19-5618.

[85] YIH W T, CHANG M W, HE X, et al. Semantic parsing via staged query graph generation: Question answering with knowledge base[C/OL]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Inter-

national Joint Conference on Natural Language Processing (Volume 1: Long Papers). Beijing, China: Association for Computational Linguistics, 2015: 1321-1331. https://aclanthology.org/P15-1128. DOI: 10.3115/v1/P15-1128.

[86] BARONI M. Linguistic generalization and compositionality in modern artificial neural networks[J]. CoRR, 2019, abs/1904.00157.

[87] MIKOLOV T, JOULIN A, BARONI M. A roadmap towards machine intelligence [C/OL]//GELBUKH A F. Computational Linguistics and Intelligent Text Processing - 17th International Conference, CICLing 2016, Konya, Turkey, April 3-9, 2016, Revised Selected Papers, Part I. Springer, 2016: 29-61. https://doi.org/10.1007/978-3-319-75477-2_2.

[88] LAKE B M, BARONI M. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks[C/OL]//Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. PMLR, 2018: 2879-2888. http://proceedings.mlr.press/v80/lake18a/lake18a.pdf.

[89] DESSÌ R, BARONI M. CNNs found to jump around more skillfully than RNNs: Compositional generalization in seq2seq convolutional networks[C/OL]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 3919-3923. DOI: 10.18653/v1/P19-1381.

[90] KEYSERS D, SCHÄRLI N, SCALES N, et al. Measuring compositional generalization: A comprehensive method on realistic data[C/OL]//8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. https://openreview.net/forum?id=SygcCnNKwr.

[91] TAI K S, SOCHER R, MANNING C D. Improved semantic representations from tree-structured long short-term memory networks[C/OL]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers. The Association for Computer Linguistics, 2015: 1556-1566. DOI: 10.3115/v1/p15-1150.

[92] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J/OL]. Neural Com-

put., 1997, 9(8):1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[93] DETHLEFS N, CUAYÁHUITL H. Combining hierarchical reinforcement learning and Bayesian networks for natural language generation in situated dialogue[C]//Proceedings of the 13th European Workshop on Natural Language Generation. Nancy, France: Association for Computational Linguistics, 2011: 110-120.

[94] JIANG Y, GU S, MURPHY K, et al. Language as an abstraction for hierarchical deep reinforcement learning[C/OL]//Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. 2019: 9414-9426. https://proceedings.neurips.cc/paper/2019/hash/0af787945872196b42c9f73ead2565c8-Abstract.html.

[95] SUTTON R S, MCALLESTER D A, SINGH S, et al. Policy gradient methods for reinforcement learning with function approximation[C/OL]//Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]. The MIT Press, 1999: 1057-1063. http://papers.nips.cc/paper/1713-policy-gradient-methods-for-reinforcement-learning-with-function-approximation.

[96] WILLIAMS R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J/OL]. Machine Learning, Volume 8, 1992:229-256. https://doi.org/10.1007/BF00992696.

[97] WEAVER L, TAO N. The optimal reward baseline for gradient-based reinforcement learning[C/OL]//UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, University of Washington, Seattle, Washington, USA, August 2-5, 2001. Morgan Kaufmann, 2001: 538-545. https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=141&proceeding_id=17.

[98] BENGIO Y, LOURADOUR J, COLLOBERT R, et al. Curriculum learning[C/OL]//Proceedings of the 26th Annual International Conference on Machine Learning. New York, NY, USA: Association for Computing Machinery, 2009: 41–48. DOI: 10.1145/1553374.1553380.

[99] CHEN X, LIU C, SONG D. Towards synthesizing complex programs from input-output examples[C/OL]//6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. https://openreview.net/forum?id=Skp1ESxRZ.

[100] HUPKES D, DANKERS V, MUL M, et al. Compositionality decomposed: How do

neural networks generalise?[J/OL]. J. Artif. Intell. Res., 2020, 67:757-795. DOI: 10.1613/jair.1.11674.

[101] LAKE B M, LINZEN T, BARONI M. Human few-shot learning of compositional instructions[C/OL]//Proceedings of the 41th Annual Meeting of the Cognitive Science Society, CogSci 2019: Creativity + Cognition + Computation, Montreal, Canada, July 24-27, 2019. cognitivesciencesociety.org, 2019: 611-617. https://mindmodeling.org/cogsci2019/papers/0123/index.html.

[102] LOULA J, BARONI M, LAKE B. Rearranging the familiar: Testing compositional generalization in recurrent networks[C/OL]//Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. Brussels, Belgium: Association for Computational Linguistics, 2018: 108-114. DOI: 10.18653/v1/W18-5413.

[103] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C/OL]// GUYON I, VON LUXBURG U, BENGIO S, et al. Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. 2017: 5998-6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[104] DEHGHANI M, GOUWS S, VINYALS O, et al. Universal transformers[C/OL]//7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. https://openreview.net/forum?id=HyzdRiR9Y7.

[105] PASZKE A, GROSS S, MASSA F, et al. Pytorch: An imperative style, high-performance deep learning library[C/OL]//Advances in Neural Information Processing Systems: volume 32. Curran Associates, Inc., 2019: 8026-8037. https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[106] ZEILER M D. AdaDelta: An adaptive learning rate method[J]. CoRR, 2012, abs/1212.5701.

[107] HAVRYLOV S, KRUSZEWSKI G, JOULIN A. Cooperative learning of disjoint syntax and semantics[C/OL]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 1118-1128. DOI: 10.18653/v1/N19-1115.

[108] ROY D. Grounding words in perception and action: computational insights[J/OL]. Trends in Cognitive Sciences, 2005, 9(8):389 - 396. http://www.sciencedirect.com/science/article/pii/S1364661305001853. DOI: https://doi.org/10.1016/j.tics.2005.06.013.

[109] ZHOU L, KALANTIDIS Y, CHEN X, et al. Grounded video description[C/OL]//IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. Computer Vision Foundation / IEEE, 2019: 6578-6587. http://openaccess.thecvf.com/content_CVPR_2019/html/Zhou_Grounded_Video_Description_CVPR_2019_paper.html. DOI: 10.1109/CVPR.2019.00674.

[110] ZHU Y, GROTH O, BERNSTEIN M S, et al. Visual7w: Grounded question answering in images[C/OL]//2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. IEEE Computer Society, 2016: 4995-5004. https://doi.org/10.1109/CVPR.2016.540.

[111] DONG Z, SUN S, LIU H, et al. Data-anonymous encoding for text-to-SQL generation [C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 5405-5414. https://aclanthology.org/D19-1543. DOI: 10.18653/v1/D19-1543.

[112] CHEN S, SAN A, LIU X, et al. A tale of two linkings: Dynamically gating between schema linking and structural linking for text-to-SQL parsing[C/OL]//Proceedings of the 28th International Conference on Computational Linguistics. Barcelona, Spain (Online): International Committee on Computational Linguistics, 2020: 2900-2912. https://aclanthology.org/2020.coling-main.260. DOI: 10.18653/v1/2020.coling-main.260.

[113] REDDY S, TÄCKSTRÖM O, COLLINS M, et al. Transforming dependency structures to logical forms for semantic parsing[J/OL]. Transactions of the Association for Computational Linguistics, 2016, 4:127-140. https://aclanthology.org/Q16-1010. DOI: 10.1162/tacl_a_00088.

[114] LEI W, WANG W, MA Z, et al. Re-examining the role of schema linking in text-to-SQL [C/OL]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics, 2020: 6943-6954. https://aclanthology.org/2020.emnlp-main.564. DOI: 10.18653/v1/2020.emnlp-main.564.

[115] SHI T, ZHAO C, BOYD-GRABER J, et al. On the potential of lexico-logical align-

ments for semantic parsing to SQL queries[C/OL]//Findings of the Association for Computational Linguistics: EMNLP 2020. Online: Association for Computational Linguistics, 2020: 1849-1864. https://aclanthology.org/2020.findings-emnlp.167. DOI: 10.18653/v1/2020.findings-emnlp.167.

[116] Samek W, Binder A, Montavon G, et al. Evaluating the visualization of what a deep neural network has learned[J/OL]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(11):2660-2673. DOI: 10.1109/TNNLS.2016.2599820.

[117] HWANG W, YIM J, PARK S, et al. A comprehensive exploration on wikisql with table-aware word contextualization[J/OL]. CoRR, 2019, abs/1902.01069. http://arxiv.org/abs/1902.01069.

[118] ARRAS L, HORN F, MONTAVON G, et al. ”what is relevant in a text document?”: An interpretable machine learning approach[J/OL]. CoRR, 2016, abs/1612.07843. http://arxiv.org/abs/1612.07843.

[119] SOROKIN D, GUREVYCH I. Mixing context granularities for improved entity linking on question answering data across entity categories[C/OL]//Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics. New Orleans, Louisiana: Association for Computational Linguistics, 2018: 65-75. https://aclanthology.org/S18-2007. DOI: 10.18653/v1/S18-2007.

[120] LIU Q, CHEN Y, CHEN B, et al. You impress me: Dialogue generation via mutual persona perception[C/OL]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, 2020: 1417-1427. https://aclanthology.org/2020.acl-main.131. DOI: 10.18653/v1/2020.acl-main.131.

[121] WOLF T, DEBUT L, SANH V, et al. Transformers: State-of-the-art natural language processing[C/OL]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Online: Association for Computational Linguistics, 2020: 38-45. https://aclanthology.org/2020.emnlp-demos.6. DOI: 10.18653/v1/2020.emnlp-demos.6.

[122] LOSHCHILOV I, HUTTER F. Decoupled weight decay regularization[C/OL]//7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. https://openreview.net/forum?id=Bkg6RiCqY7.

[123] LI B Z, MIN S, IYER S, et al. Efficient one-pass end-to-end entity linking for questions

[C/OL]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics, 2020: 6433-6441. https://aclanthology.org/2020.emnlp-main.522. DOI: 10.18653/v1/2020.emnlp-main.522.

[124] LIN X V, SOCHER R, XIONG C. Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing[C/OL]//Findings of the Association for Computational Linguistics: EMNLP 2020. Online: Association for Computational Linguistics, 2020: 4870-4888. https://aclanthology.org/2020.findings-emnlp.438. DOI: 10.18653/v1/2020.findings-emnlp.438.

[125] XU S, SEMNANI S, CAMPAGNA G, et al. AutoQA: From databases to QA semantic parsers with only synthetic training data[C/OL]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics, 2020: 422-434. https://aclanthology.org/2020.emnlp-main.31. DOI: 10.18653/v1/2020.emnlp-main.31.

[126] ARTZI Y, ZETTLEMOYER L. Weakly supervised learning of semantic parsers for mapping instructions to actions[J/OL]. Transactions of the Association for Computational Linguistics, 2013, 1:49-62. https://aclanthology.org/Q13-1005. DOI: 10.1162/tacl_a_00209.

[127] GOLDMAN O, LATCINNIK V, NAVE E, et al. Weakly supervised semantic parsing with abstract examples[C/OL]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, 2018: 1809-1819. https://aclanthology.org/P18-1168. DOI: 10.18653/v1/P18-1168.

[128] DASIGI P, GARDNER M, MURTY S, et al. Iterative search for weakly supervised semantic parsing[C/OL]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 2669-2680. https://aclanthology.org/N19-1273. DOI: 10.18653/v1/N19-1273.

[129] WANG B, TITOV I, LAPATA M. Learning semantic parsers from denotations with latent structured alignments and abstract programs[C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint

Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 3774-3785. https://aclanthology.org/D19-1391. DOI: 10.18653/v1/D19-1391.

[130] WANG B, LAPATA M, TITOV I. Learning from executions for semantic parsing [C/OL]//Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Online: Association for Computational Linguistics, 2021: 2747-2759. https://aclanthology.org/2021.naacl-main.219. DOI: 10.18653/v1/2021.naacl-main.219.

[131] CHOI E, HE H, IYYER M, et al. QuAC: Question answering in context[C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 2174-2184. https://aclanthology.org/D18-1241. DOI: 10.18653/v1/D18-1241.

[132] BAO H, DONG L, WEI F, et al. Unilmv2: Pseudo-masked language models for unified language model pre-training[C/OL]//Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. PMLR, 2020: 642-652. http://proceedings.mlr.press/v119/bao20a.html.

[133] LEWIS M, LIU Y, GOYAL N, et al. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension[C/OL]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, 2020: 7871-7880. https://aclanthology.org/2020.acl-main.703. DOI: 10.18653/v1/2020.acl-main.703.

[134] HENDRYCKS D, GIMPEL K. Bridging nonlinearities and stochastic regularizers with gaussian error linear units[J/OL]. CoRR, 2016, abs/1606.08415. http://arxiv.org/abs/1606.08415.

[135] CHEN W, WANG H, CHEN J, et al. Tabfact: A large-scale dataset for table-based fact verification[C/OL]//8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. https://openreview.net/forum?id=rkeJRhNYDH.

[136] WANG B, YIN W, LIN X V, et al. Learning to synthesize data for semantic parsing [C/OL]//Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Online: Association for Computational Linguistics, 2021: 2760-2766. https://aclanthology.org/202

1.naacl-main.220. DOI: 10.18653/v1/2021.naacl-main.220.

[137] ZHONG V, LEWIS M, WANG S I, et al. Grounded adaptation for zero-shot executable semantic parsing[C/OL]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics, 2020: 6869-6882. https://aclanthology.org/2020.emnlp-main.558. DOI: 10.18653/v1/2020.emnlp-main.558.

[138] OTT M, EDUNOV S, BAEVSKI A, et al. fairseq: A fast, extensible toolkit for sequence modeling[C/OL]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 48-53. https://aclanthology.org/N19-4009. DOI: 10.18653/v1/N19-4009.

[139] WANG C, TATWAWADI K, BROCKSCHMIDT M, et al. Robust text-to-sql generation with execution-guided decoding[J]. CoRR, 2018, abs/1807.03100.

[140] XUAN K, WANG Y, WANG Y, et al. Sead: End-to-end text-to-sql generation with schema-aware denoising[J/OL]. CoRR, 2021, abs/2105.07911. https://arxiv.org/abs/2105.07911.

[141] NEELAKANTAN A, LE Q V, SUTSKEVER I. Neural programmer: Inducing latent programs with gradient descent[C/OL]//4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. 2016. http://arxiv.org/abs/1511.04834.

[142] ZHANG Y, PASUPAT P, LIANG P. Macro grammars and holistic triggering for efficient semantic parsing[C/OL]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, 2017: 1214-1223. https://aclanthology.org/D17-1125. DOI: 10.18653/v1/D17-1125.

[143] QIN B, WANG L, HUI B, et al. SDCUP: schema dependency-enhanced curriculum pre-training for table semantic parsing[J/OL]. CoRR, 2021, abs/2111.09486. https://arxiv.org/abs/2111.09486.

[144] IYYER M, YIH W, CHANG M. Search-based neural structured learning for sequential question answering[C/OL]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers. Association for Computational Linguistics, 2017: 1821-1831.

https://doi.org/10.18653/v1/P17-1167.

[145] SUN Y, TANG D, XU J, et al. Knowledge-aware conversational semantic parsing over web tables[C/OL]//Natural Language Processing and Chinese Computing - 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9-14, 2019, Proceedings, Part I. Springer, 2019: 827-839. https://doi.org/10.1007/978-3-030-32233-5_64.

[146] MUELLER T, PICCINNO F, SHAW P, et al. Answering conversational questions on structured data without logical forms[C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 5902-5910. https://aclanthology.org/D19-1603. DOI: 10.18653/v1/D19-1603.

[147] YU T, ZHANG R, POLOZOV A, et al. Score: Pre-training for context representation in conversational semantic parsing[C/OL]//9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. https://openreview.net/forum?id=oyZxhRI2RiE.

[148] EISENSCHLOS J, KRICHENE S, MÜLLER T. Understanding tables with intermediate pre-training[C/OL]//Findings of the Association for Computational Linguistics: EMNLP 2020. Online: Association for Computational Linguistics, 2020: 281-296. https://aclanthology.org/2020.findings-emnlp.27. DOI: 10.18653/v1/2020.findings-emnlp.27.

[149] ZHANG S, MA X, RUDINGER R, et al. Cross-lingual decompositional semantic parsing [C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 1664-1675. https://aclanthology.org/D18-1194. DOI: 10.18653/v1/D18-1194.

[150] CHEN X, GONG L, CHEUNG A, et al. PlotCoder: Hierarchical decoding for synthesizing visualization code in programmatic context[C/OL]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Online: Association for Computational Linguistics, 2021: 2169-2181. https://aclanthology.org/2021.acl-long.169. DOI: 10.18653/v1/2021.acl-long.169.

[151] BERTOMEU N, USZKOREIT H, FRANK A, et al. Contextual phenomena and thematic relations in database QA dialogues: results from a Wizard-of-Oz experiment[C/OL]//Proceedings of the Interactive Question Answering Workshop at HLT-NAACL 2006.

New York, NY, USA: Association for Computational Linguistics, 2006: 1-8. https://aclanthology.org/W06-3001.

[152] SU H, SHEN X, ZHANG R, et al. Improving multi-turn dialogue modelling with utterance ReWriter[C/OL]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 22-31. https://aclanthology.org/P19-1003. DOI: 10.18653/v1/P19-1003.

[153] ELGOHARY A, PESKOV D, BOYD-GRABER J. Can you unpack that? learning to rewrite questions-in-context[C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 5918-5924. https://aclanthology.org/D19-1605. DOI: 10.18653/v1/D19-1605.

[154] PAPINENI K, ROUKOS S, WARD T, et al. Bleu: a method for automatic evaluation of machine translation[C/OL]//Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, 2002: 311-318. https://aclanthology.org/P02-1040. DOI: 10.3115/1073083.1073135.

[155] SEO M J, KEMBHAVI A, FARHADI A, et al. Bidirectional attention flow for machine comprehension[C/OL]//5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net, 2017. https://openreview.net/forum?id=HJ0UKP9ge.

[156] KIM Y. Convolutional neural networks for sentence classification[C/OL]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, October 25-29, 2014. 2014: 1746-1751. http://aclweb.org/anthology/D/D14/D14-1181.pdf.

[157] PENNINGTON J, SOCHER R, MANNING C D. GloVe: Global vectors for word representation[C/OL]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, October 25-29, 2014. 2014: 1532-1543. http://aclweb.org/anthology/D/D14/D14-1162.pdf.

[158] WANG W, CHANG B. Graph-based dependency parsing with bidirectional LSTM [C/OL]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Com-

putational Linguistics, 2016: 2306-2315. https://aclanthology.org/P16-1218. DOI: 10.18653/v1/P16-1218.

[159] LEE K, HE L, LEWIS M, et al. End-to-end neural coreference resolution[C/OL]// Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, 2017: 188-197. https://aclanthology.org/D17-1018. DOI: 10.18653/v1/D17-1018.

[160] GARDNER M, GRUS J, NEUMANN M, et al. AllenNLP: A deep semantic natural language processing platform[C/OL]//Proceedings of Workshop for NLP Open Source Software (NLP-OSS). Melbourne, Australia: Association for Computational Linguistics, 2018: 1-6. https://aclanthology.org/W18-2501. DOI: 10.18653/v1/W18-2501.

[161] KINGMA D P, BA J. Adam: A method for stochastic optimization[C/OL]//3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015. http://arxiv.org/abs/1412.6980.

[162] PAN Z, BAI K, WANG Y, et al. Improving open-domain dialogue systems via multi-turn incomplete utterance restoration[C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 1824-1833. https://aclanthology.org/D19-1191. DOI: 10.18653/v1/D19-1191.

[163] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners[EB/OL]. 2019. https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[164] YU S, LIU J, YANG J, et al. Few-shot generative conversational query rewriting[C/OL]// Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020. ACM, 2020: 1933-1936. https://doi.org/10.1145/3397271.3401323.

[165] KUMAR V, JOSHI S. Non-sentential question resolution using sequence to sequence learning[C/OL]//Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. Osaka, Japan: The COLING 2016 Organizing Committee, 2016: 2022-2031. https://aclanthology.org/C16-1190.

攻读博士学位期间取得的研究成果

已发表或已录用的论文

[1] **Qian Liu**, Bei Chen, Jian-Guang Lou, Ge Jin, Dongmei Zhang. FANDA: A Novel Approach to Perform Follow-up Query Analysis. In *Proceedings of the 33th AAAI Conference on Artificial Intelligence(AAAI-2019)*, AAAI Press, Hawaii, USA, 2019, pages 6770–6777.（长文，已发表，中国计算机学会推荐的 A 类国际会议）

[2] **Qian Liu**, Bei Chen, Haoyan Liu, Lei Fang, Jian-Guang Lou, Bin Zhou, Dongmei Zhang. A Split-and-Recombine Approach for Follow-up Query Analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing(EMNLP-2019)*, Association for Computational Linguistics, Hong Kong, China, 2019, pages 5315–5325.（长文，已发表，中国计算机学会推荐的 B 类国际会议）

[3] **Qian Liu**, Yihong Chen, Bei Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, Dongmei Zhang. You Impress Me: Dialogue Generation via Mutual Persona Perception. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics(ACL-2020)*, Association for Computational Linguistics, Online, 2020, pages 1417–1427.（长文，已发表，中国计算机学会推荐的 A 类国际会议）

[4] **Qian Liu**, Bei Chen, Jiaqi Guo, Jian-Guang Lou, Bin Zhou and Dongmei Zhang. How Far are We from Effective Context Modeling? An Exploratory Study on Semantic Parsing in Context. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence(IJCAI-2020)*, International Joint Conferences on Artificial Intelligence Organization, Online, 2020, pages 3580–3586.（长文，已发表，中国计算机学会推荐的 A 类国际会议）

[5] **Qian Liu**, Bei Chen, Jian-Guang Lou, Bin Zhou and Dongmei Zhang. Incomplete Utterance Rewriting as Semantic Segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing(EMNLP-2020)*, Association for Computational Linguistics, Online, 2020, pages 2846-2857.（长文，已发表，中国计算机学会推荐的 B 类国际会议）

[6] **Qian Liu**, Shengnan An, Jian-Guang Lou, Bei Chen, Zeqi Lin, Yan Gao, Bin Zhou, Nanning Zheng, Dongmei Zhang. Compositional Generalization by Learning Analytical Expressions. In *Proceedings of Advances in Neural Information Processing Systems*

33: Annual Conference on Neural Information Processing Systems 2020(NeurIPS-2020), Curran Associates Inc., Online, 2020, pages 11416–11427.（长文，已发表，中国计算机学会推荐的 A 类国际会议）

[7] **Qian Liu**, Dejian Yang, Jiahui Zhang, Jiaqi Guo, Bin Zhou, Jian-Guang Lou. Awakening Latent Grounding from Pretrained Language Models for Semantic Parsing. In *Findings of the Association for Computational Linguistics: ACL 2021(ACL-2021 Findings)*, Association for Computational Linguistics, Online, 2021, pages 1174-1189.（长文，已发表）

[8] **Qian Liu**, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, Jian-Guang Lou. TAPEX: Table Pre-training via Learning a Neural SQL Executor. In *Proceedings of the 10th International Conference on Learning Representations(ICLR-2022)*, 2022.（长文，已录用）

[9] Jiaqi Guo, **Qian Liu**, Jian-Guang Lou, Zhenwen Li, Xueqing Liu, Tao Xie and Ting Liu. Benchmarking Meaning Representations in Neural Semantic Parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing(EMNLP-2020)*, Association for Computational Linguistics, Online, 2020, pages 1520–1540.（长文，已发表，中国计算机学会推荐的 B 类国际会议）

[10] Shuang Chen, **Qian Liu**, Zhiwei Yu, Chin-Yew Lin, Jian-Guang Lou, Feng Jiang. ReTraCk: A Flexible and Efficient Framework for Knowledge Base Question Answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics: System Demonstrations(ACL-2021 Demo)*, Association for Computational Linguistics, Online, 2021, pages 325-336.（长文，已发表）

[11] Haoyan Liu, Lei Fang, **Qian Liu**, Bei Chen, Jian-Guang Lou, Zhoujun Li. Leveraging Adjective-noun Phrasing Knowledge for Comparison Relation Prediction in Text-to-SQL. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing(EMNLP-2019)*, Association for Computational Linguistics, Hong Kong, China, 2019, pages 3513–3518.（长文，已发表，中国计算机学会推荐的 B 类国际会议）

[12] Yuntao Li, Bei Chen, **Qian Liu**, Yan Gao, Jian-Guang Lou, Yan Zhang and Dongmei Zhang. “What Do You Mean by That?” A Parser-Independent Interactive Approach for Enhancing Text-to-SQL. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing(EMNLP-2020)*, Association for Computational Linguistics, Online, 2020, pages 6913–6922.（长文，已发表，中国计算机学会推荐的 B 类

国际会议）

[13] Yuntao Li, Bei Chen, **Qian Liu**, Yan Gao, Jiang-Guang Lou, Yan Zhang, Dongmei Zhang. Keep the Structure: A Latent Shift-Reduce Parser for Semantic Parsing. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence(IJCAI-2021)*, International Joint Conferences on Artificial Intelligence Organization, Online, 2021, pages 3864–3870.（长文，已发表，中国计算机学会推荐的 A 类国际会议）

[14] Yu Zeng, Yan Gao, Jiaqi Guo, Bei Chen, **Qian Liu**, Jian-Guang Lou, Fei Teng, Dongmei Zhang. RECPARSER: A Recursive Semantic Parsing Framework for Text-to-SQL Task. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence(IJCAI-2020)*, International Joint Conferences on Artificial Intelligence Organization, Online, 2020, pages 3644–3650.（长文，已发表，中国计算机学会推荐的 A 类国际会议）

[15] Jiaqi Guo, Ziliang Si, Yu Wang, **Qian Liu**, Ming Fan, Jian-Guang Lou, Zijiang Yang, Ting Liu. Chase: A Large-Scale and Pragmatic Chinese Dataset for Cross-Database Context-Dependent Text-to-SQL. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics(ACL-2021)*, Association for Computational Linguistics, Online, 2021, pages 2316–2331.（长文，已发表，中国计算机学会推荐的 A 类国际会议）

在读期间所获奖励

[1] 2019 年度北京航空航天大学博士研究生国家奖学金

[2] 2020 年度百度奖学金提名奖

[3] 2021 年度北京航空航天大学博士研究生国家奖学金

[4] 2021 年度北京航空航天大学博士研究生卓越学术基金

致谢

再提笔写下致谢，时光匆匆已五年。仿佛昨天我刚完成本科毕业典礼的拨穗仪式，今天我的漫漫科研路就要随着博士论文的即将完成告一段落。读博这五年是枯燥乏味的，有过书海遨游硬啃数值分析的辛酸苦楚，也有因疫情被迫居家科研的孤独时光。读博这五年又是多姿多彩的，有过参加国际会议向外国友人介绍成果的美好记忆，也有与同行的伙伴建立的深厚友情。在博士论文即将完成之际，在此我要向一直以来指导、关心和帮助过我的各位老师、同学、朋友和亲人致以最诚挚的谢意。

首先，衷心感谢我的导师赵沁平教授与微软亚洲研究院的楼建光首席研究员。回想起五年前，我刚加入北航计算机学院与微软亚洲研究院的联合培养博士项目，在那之前我从未接触过人工智能领域，不知道神经网络为何物，更是没有看过一篇相关论文。是两位老师高瞻远瞩建议我开展自然语言语义解析的研究课题，也是两位老师指导我一步步从科研小白成长为可以独当一面的研究者。两位老师是我科研道路上的师长，亦是我生活上的朋友。我记得赵老师听我讲最近生活时的慈祥笑容，也记得在楼老师在我心情沮丧时的体贴宽慰。能加入联合培养博士项目是我人生中的一大幸事，两位老师带给我的学术启迪也将成为我人生中最珍贵的财富！在此向两位老师致以最崇高的敬意！

衷心感谢周彬老师与微软亚洲研究院的陈蓓高级研究员。如果说赵老师和楼老师是帮助我指明了正确的科研方向，那周老师和蓓姐就是在科研漫漫旅途上一点点地教我从爬行到学会走路，从小跑到学会快跑。仍记得第一次撰写论文时蓓姐一字一句地帮我重写了整篇论文，让我从修改中学会了科学论文正确的逻辑结构。也记得周老师不厌其烦地为我的博士开题报告提意见，让我在一次次的讨论中梳理清楚了博士论文的结构，也有幸获得了优异的开题成绩。你们的建议和鼓励让我得以克服困难，不断前行。

感谢在我求学期间曾指导过我的老师和研究员们，包括林泽琦、杨德剑、Weizhu Chen、Morteza Ziyadi、余智薇、付强、邹欣、胡晓武、孙丽君、马歆、房磊、高妍、孙诗昭、Börje F. Karlsson、肖岩、林梓佳、曹涌、聂再清、郭沐、张冬梅老师等，每一次和老师们的讨论交流都能让我对前沿研究有更加深入的了解，老师们对我的关心和体贴也永不会忘！

感谢虚拟国家重点实验室的同门师兄弟姐妹们，包括卢飞翔、金详凯、杨明佳、周生迪、石亚豪、刘宗岱、杨新航、杨义轩、彭昊天、张松、苗荟等，每周六开组会时与你们的交流让我获益良多。感谢微软亚洲研究院一起实习的小伙伴们，包括陈双、鲍航波、候宇泰、张攀、郭昊翔、骆煦芳、吴双志、任硕、蒋墨岚、张醒之、姚凯、施韩原、

李讴邑、唐可文、刘昊岩、杨泽、范志鹏、段旭光、骆梁宸、王本亮、江政宝、陈三星、陈艺虹、刘子初、赵田阳、朱钰颖、徐思成、武千惠、赵冬迪、詹泽诚、郭家琦、林禹臣、闫韶光、王利兴、闫坤、苑浩、谢培卓、王运里、丁嘉榆、曹雁彬、陈永强、李云涛、曾俞、安晟南、林宇桐、郭一诺、宫永顺、钱纯瑶、王佳琪、胡扬、刘晨瑶、孔文苑、朱昆睿、孙占辰、张家辉、郭茁宁、刘沛成、窦隆绪、马婷婷、王冰、皮鑫雨、覃立波、李一飞、施琦、咎道广、张子潇、陈志、钟宛君、郭达雅、黄俊杰、石恩升、姚朱亮、赵健安、王怡琦、王程一、刘泽、毛海涛、王浩学、徐嘉梁、徐啸等，与你们一起赶论文、聚餐、八卦的时光将是我一生中最宝贵的回忆。

感谢百忙之中评阅我毕业论文并给出宝贵意见的评审老师们，向你们表达我最诚挚的谢意。

最后，特别感谢我的父母，是你们的全力支持让我得以在求学期间心无旁骛地选择自己想走的路。从选择计算机方向到攻读博士，你们对我无条件的爱和开明的家庭氛围是让我走到今天的最大动力。尤其感谢我的女友，你的安慰是治愈我伤口的良药，你的体贴让我时刻感受到被爱，谢谢你多年来无私的陪伴、支持与鼓励！

作者简介

刘乾，男，1996 年 2 月 19 日出生于山西省忻州市，北京航空航天大学博士研究生。

2013 年 9 月考入北京航空航天大学计算机系计算机科学与技术专业，2017 年 7 月本科毕业并获得工学学士学位。

2017 年 9 月保送进入北京航空航天大学计算机系攻读工学博士学位至今。