arXiv:2301.00399v1 [cs.CL] 1 Jan 2023

语义算子预测与应用

Farshad Noravesh*

2023年1月3日

摘要

本文简要介绍了语义解析面临的挑战，并使用带注意力机制的序列到序列模型实现了语义解析中的 QDMR形式体系。不过，为了尽可能简化并加快训练，同时避免维数灾难和过拟合，该模型仅使用词性（POS）来表示句子中的词。本文还展示了如何将语义算子预测与 CopyNet 模型或递归神经网络模型等其他模型结合。

1 引言

近年来，语义解析与问答逐渐紧密结合，原因有很多，例如技术原因之一——远程监督：与树库相比，为问答创建数据集要简单得多。另一种弱监督方法是，仅将树中最高节点的逻辑形式视为唯一的监督来源，正如（Herzig & Berant 2021）所做的那样。

据观察，当前最先进的基于知识的问答（KBQA）模型主要采用语义解析生成逻辑形式，以便直接在这些知识图谱上执行，如（Gu et al. 2022）、（Gu et al. 2021）和（Berant et al. 2013）中所述；或者将语义解析任务与知识库交互分离开来，这一方法由（Ravishankar et al. 2021）提出。

*电子邮件：noraveshfarshad@gmail.com

传统上，实体链接（查找给定问题中提及的实体）被视为语义解析的子问题，并假定会预先完成；而（Krishnamurthy et al. 2017）则将实体链接与语义解析结合起来。语义解析有多种形式体系，例如抽象意义表示（AMR）、话语表示结构（DRS）、结构化查询语言（例如 SQL）和 λ演算。（Kapanipathi et al. 2020）将 AMR 用于基于知识的问答，它整合了多个可复用模块，如语义解析器、实体和关系链接器以及神经符号推理器。除了基于知识的问答，数据来源也可以是表格。在这类任务中，答案格式可以是短语实体，也可以是自由形式文本，例如（Nan et al. 2022）中的 FeTaQA 数据集。

语义解析是人工智能中许多具有挑战性问题的基础模块，例如对话系统、问答，也能增强基于对话式人工智能、阅读理解和故事生成的技术。传统的问答方法，如（Zhou et al. 2018）、（Shi et al. 2021）、（Zhang et al. 2022）和（Ren et al. 2021），并未利用语义解析，因此其方法的可解释性较差，也难以泛化。对于开放域问答，这一问题尤为突出，例如（Sun et al. 2018）和（Sun et al. 2019）。一般而言，解析有三种方法，即自顶向下、自底向上和混合方法。尽管（Yang & Tu 2022）提出的自底向上成分句法分析效率很高，但其可扩展性不足，因为从众包的角度来看，获取这些复杂的序列标注成本高昂。这表明可以采用两种范式来处理这一问题。第一种思路是使用远程监督，将问答问题中的误差反向传播到语义解析中。（Wolfson et al. 2020）提出的第二种思路则是创建一个中间层，使众包成本更低、可扩展性更强。第 2 节说明了为何某些方法缺乏可扩展性，且在实践中实施成本高昂。第 3 节展示了 QDMR形式体系如何有助于构建任何可扩展的语义解析算法。第 4 节提出了一种模型。本文的主要贡献之一是强调词典式对齐和解耦信息处理。近年来，人们开始关注利用语义标注进行语义解析，例如（Zheng & Lapata 2020）所做的那样：首先将语义标签视为潜变量，然后使用这些语义标签序列来学习逻辑形式，例如 SQL 类型或 λ演算形式。训练可以单独进行，也可以采用端到端方式联合进行。

2 昂贵的语义解析

构建树库是当前语义解析方法面临的首要挑战。与句法解析中的树库构建相比，这一问题更加突出，因为除了歧义之外，众包标注人员的成本也更高，每个句子的标注耗时更长，而且标注人员还应熟悉组合范畴语法（CCG）、λ演算、类型提升以及组合范畴语法（CCG）中的组合等复杂形式体系。像（Yang & Tu 2022）那样使用指针网络，需要昂贵的众包——

```
question_id,question_text,decoder_input_span,decoder_input_label,decoder_output_pointed,decoder_output_label
ACADEMIC_train_0,return me the homepage of PVLDB .,start_from start_to 0 2 2 4,start span select,2 4 7,span
select filter
ACADEMIC_train_1,return me the homepage of H.V.Jagadish .,start_from start_to 0 2 2 4,start span select,2 4
7,span select filter
ACADEMIC_train_10,return me the number of references of making database systems usable .,start_from start_to
0 2 2 5 5 6 6 12 12 2,start span skip select filter skip,2 5 6 12 2 5,span skip select filter skip aggregate
ACADEMIC_train_100,return me the number of papers in VLDB conference containing keyword Information
Retrieval .,start_from start_to 0 2 2 4 4 6 6 9 9 14 14 2,start span skip select filter filter skip,2 4 6 9
14 2 4,span skip select filter filter skip aggregate
ACADEMIC_train_101,return me the number of authors who have papers containing keyword Relational Database
.,start_from start_to 0 2 2 8 8 9 9 14 14 4 4 6 6 2,start span skip select filter skip project skip,2 8 9 14
4 6 2 4,span skip select filter skip project skip aggregate
```

图 1：昂贵的数据集

数据来源。虽然这种方法曾用于句法分析，但也可以采用相同的方法，通过创建一个如图 1 所示的数据集，将其应用于语义解析；不过在实际中，这样做的成本很高。图 2 展示了这样一个假想的高成本模型。

QDMR 数据集（Wolfson et al. 2020）中的典型句子可以使用语义算子进行解析，如图 3 所示。例如，当解码器位于光标 6 时，它指向边界 12（of making database systems usable），其对应的语义算子标签为“过滤”。

近年来，语义解析建模开始采用分阶段的方式（如 Dong & La- pata 2018），逐步处理输入话语，先从粗略层面着手，再深入细节。因此，他们首先生成一个对其含义的粗略草图，其中略去了变量名称和参数。然后，再根据输入话语中的细节，适当地补全草图中缺失的信息。另一个分阶段处理的例子是 Wolfson et al. 2020，他们从不同角度解决这一问题：创建一个中间层，使众包标注更容易进行，且无需具备复杂逻辑形式、λ演算和 CCG 方面的专业知识。下一节将展示，BREAK 数据集中的这种分阶段机制如何加快标注过程，并以相对低廉的成本创建大型数据集。

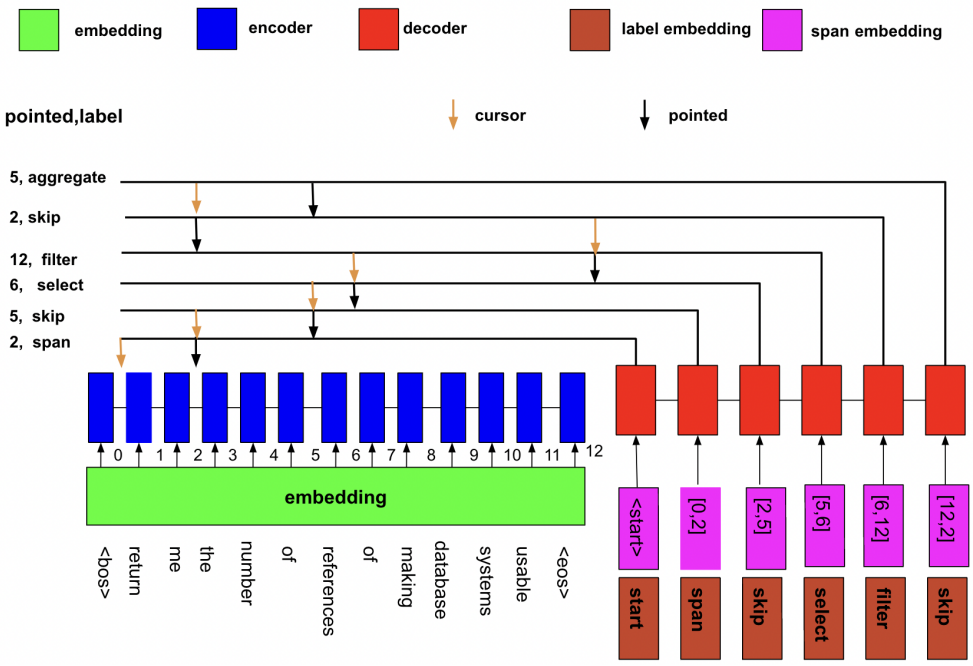

图 2：昂贵模型

3 问题分解

许多问答方法（如 Yavuz 等人，2022）并未采用语义解析。其中一个原因是，从数据集开发的角度来看，当前的语义解析形式体系成本高昂，而且难以实现。正如 Min 等人，2019、Perez 等人，2020 所提到的，问题分解研究正迅速发展。在语义解析中，一种有效的远程监督方法是利用图 4 中由黄金答案产生的误差进行误差反向传播。因此，这种问题分解中各个子问题的语义逻辑规则被视为潜变量，不直接参与监督。问题分解极具启发性，Wolfson et al. 2020 因此推出了 BREAK 数据集，并定义了 QDMR（问题分解含义表示）。该数据集包含超过 83K 对问题及其对应的 QDMR，也可用于开放域问答。BREAK 数据集包含十三种操作符，其中五种如图 5 所示。通过将 Gu 等人，2016 提出的 CopyNet 用于 BREAK 数据集，这一语义解析问题可以视为机器翻译问题。尽管这个问题似乎已经得到更多.....

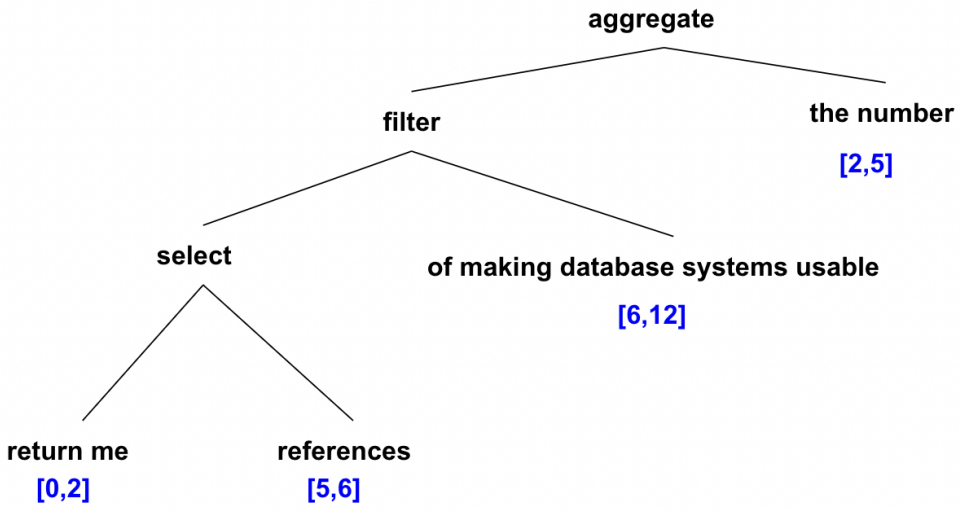

图 3：高昂的解析成本

准确率超过 70%，但这种建模方法缺乏可解释性，因此，要实现更好的泛化，组合性是必要的。

4 模型

使用词性标签而非词语词元的动机在于降低模型的复杂度。还有一种更好的表示方式，即词性，它在（Abzianidze & Bos 2017）中被称为“通用语义标签”，因为它兼具词性标签和命名实体类别的语义优点，但本文并未采用这种表示方式。

使用 word2vec 为句子中的词分配高维向量，同时为每个词性标签分配小维度向量，就足以充分表示该标签，因此无需用向量表示词语，从而减轻维数灾难。正如下一节的实验将展示的那样，每个标签只需一个二维或三维向量，就能形成足够丰富的表示。词性标签的嵌入与语义算子共同学习。因此，这种做法能在模型复杂度与预测误差之间取得良好权衡，同时避免过拟合。

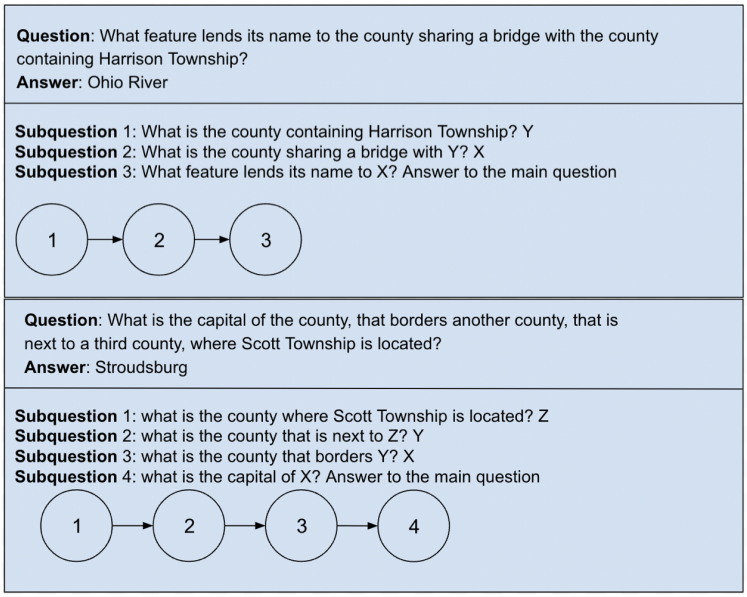

图 4：问题分解

Operator	Template / Signature	Question	Decomposition
Select	Return [entities] $w \rightarrow S_e$	How many touchdowns were scored overall?	1. **Return touchdowns** 2. Return the number of #1
Filter	Return [ref] [condition] $S_o, w \rightarrow S_o$	I would like a flight from Toronto to San Diego please.	1. Return flights **2. Return #1 from Toronto** **3. Return #2 to San Diego**
Project	Return [relation] of [ref] $w, S_e \rightarrow S_o$	Who is the head coach of the Los Angeles Lakers?	1. Return the Los Angeles Lakers **2. Return the head coach of #1**
Aggregate	Return [aggregate] of [ref] $w_{agg}, S_o \rightarrow n$	How many states border Colorado?	1. Return Colorado 2. Return border states of #1 **3. Return the number of #2**
Group	Return [aggregate] [ref1] for each [ref2] $w_{agg}, S_o, S_e \rightarrow S_n$	How many female students are there in each club?	1. Return clubs 2. Return female students of #1 **3. Return the number of #2 for each #1**

图 5：BREAK

4.1 模型概述

所提出的模型类似于带注意力机制的标准编码器-解码器网络（如 Bahdanau 等人，2014）。 x_t 是句子的输入词性标注序列， y_t是生成的语义算子序列。编码器和解码器均采用门控循环单元（GRU），因为它们的运行速度相对快于 LSTM，且参数较少。编码器的隐藏向量如下：

$$h(t) = f(x_t, h_{t-1}) \tag{4.1}$$

每个语义算子序列的概率为：

$$
\begin{aligned}
p(y) &= \prod_{t=1}^{T} p(y_t|y_1, \ldots, y_{t-1}, x) \\
p(y_t|y_1, \ldots, y_{t-1}, x) &= g(y_{t-1}, s_t, c_t)
\end{aligned}
\tag{4.2}
$$

其中， s_t 是解码器在时间步 t 的隐藏状态， c_t 是时间步 t 的上下文。Bahdanau 注意力（Bahdanau 等人，2014）的权重用于关注不同的词性标签，以使语义算子与词性标签对齐。因此，上下文向量 c_t 表示为：

$$
c_t = \sum_{j=1}^{T_x} \alpha_{tj} h_j \tag{4.3}
$$

其中，权重 α_{tj} 如下：

$$
\alpha_{tj} = \frac{\exp(a(s_{t-1}, h_j))}{\sum_{k=1}^{T_x} \exp(a(s_{t-1}, h_k))} \tag{4.4}
$$

对齐模型是以下单层感知机：

$$
a(s_{t-1}, h_j) = v_a^T \tanh(W_a s_{t-1} + U_a h_j) \tag{4.5}
$$

其中， v_a， W_a， U_a 是需要训练的权重。最后，公式 4.2 中的函数 g 用于预测语义算子，其模型简单地表示为一个线性层，该层使用权重 W_{op} 作用于前一个预测的语义算子、时间步 t 的上下文以及解码器前一个隐藏状态的拼接。因此：

$$
g(y_{t-1}, s_t, c_t) = W_{op}(y_{t-1}; c_t; s_{t-1}) \tag{4.6}
$$

4.2 实验

图 6 展示了第一次实验的结果。在这种情况下，编码器的嵌入维度和隐藏状态维度均保持为较小的 3，因为无需增加模型复杂度。训练过程中使用了教师强制来加快训练速度。在图7所示的第二次实验中，编码器和解码器的嵌入维度及隐藏状态维度均有所增加，但准确率并未显著提升，这再次表明，表示词性标签并不需要较大的向量。语义算子仅有 13 个，因此嵌入的复杂度

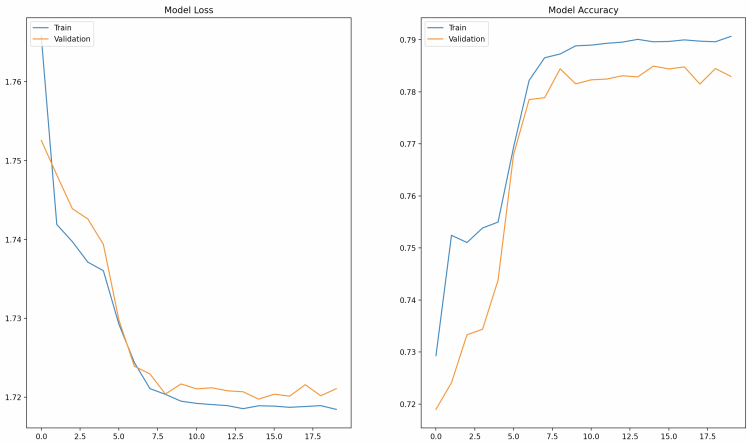

图 6：实验1

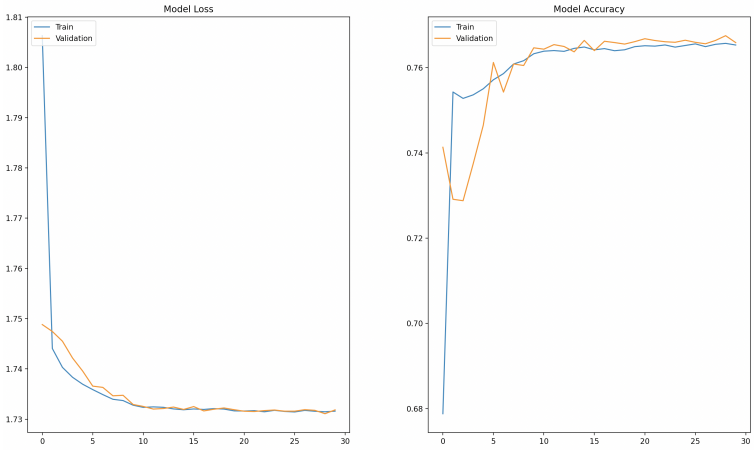

图7：实验2

解码器的隐藏状态也保持较小，以降低模型复杂度。两次实验中均使用调度器，每 10 个训练轮数降低一次学习率。原本可以使用针对词性标签的预训练 word2vec 模型，但本文将词性标签和语义算子的嵌入与模型联合学习。

表 1：实验

ex	opt	训练轮数	起始 lr	批次 size	教师强制	编码器 emb-Dim	编码器 hid-Dim	解码器 emb-Dim	解码器 hid-Dim
1	Adam	20	1e-3	10	0.5	3	3	3	3
2	SGD	30	1e-2	5	0.5	5	10	4	12

5 应用

第 4 节介绍了该模型的两种应用。第一种应用展示了如何通过添加语义算子作为额外特征，将语义算子预测用于构建表达能力更强的 CopyNet 模型。第二种应用展示了如何在基于图的方法中使用这些操作符，以获得表达能力更强的评分；该方法可以通过最大间隔损失进行训练，甚至可以使用简单的交叉熵损失进行训练，正如 Pasupat 等人，2019 中所采用的那样。

5.1 以条件信息增强 CopyNet

利用 CopyNet 的思想（Gu 等人，2016）对 QDMR 进行监督学习十分直接，许多研究者都采用了这种方法。本文中的语义算子预测可作为 CopyNet 的额外特征，并可通过多种方式实现。Gu 等人，2016 对 CopyNet 的原始公式采用以下概率来生成目标词 y_t:

$$p(y_t|s_t, y_{t-1}, c_t, M) = p(y_t, g|s_t, y_{t-1}, c_t, M) + p(y_t, c|s_t, y_{t-1}, c_t, M) \quad (5.1)$$

其中， $M = \{h_1, \ldots, h_{T_S}\}$，g 表示生成器模式，c 表示复制模式， c_t表示时间步 t 的上下文。现在，可以使用第 4 节模型的结果，引入一个额外的表达性特征——语义算子 $alsop_t$ ，作为解码器时间步 t 的条件。因此，

$$p(y_t|s_t, y_{t-1}, c_t, M, alsop_t) = p(y_t, g|s_t, y_{t-1}, c_t, M, alsop_t) + p(y_t, c|s_t, y_{t-1}, c_t, M, alsop_t) \quad (5.2)$$

有许多方法可以对 $p(y_t, g|s_t, y_{t-1}, c_t, M, alsop_t)$ 建模，但新的问题是如何将 $sop_{t'}$ 调用的语义算子预测与解码器对齐

时间步来对 $alsop_t$ 建模。请注意， $sop_{t'}$ 中的 t' 指的是用于算子预测的解码器，而 $alsop_t$ 中的 t 指的是 CopyNet 模型中解码器的时间步，两者应该对齐。一个思路是定义两个动作，即“使用_当前”动作和“使用_下一个”动作。可以通过 softmax 函数后接多层感知机（MLP）来预测这两个标签。第一个标签“使用_当前”通知解码器直接使用当前的语义算子预测结果，此时两者仍然对齐。第二个标签“使用_下一个”表示发生了错位，因此必须将指针向前移动一步，使两个序列对齐。因此，使用以下多层感知机对其建模：

$$action(t) = softmax(MLP(y_{t-1}, sop_{t'})) \tag{5.3}$$

其中，公式 5.3 中的 y_{t-1} 表明，该操作对 CoyNet 解码器模型生成的词语非常敏感。它还对潜变量模型在时间步 t' 的算子预测值敏感。现在，$alsop_t$ 可通过以下关系获得：

$$alsop_{t+1} = \left\{ \begin{array}{ll} sop_{t'}, & \text{if action is current} \\ sop_{t'+1}, & \text{if action is next} \end{array} \right\}$$

最简单的想法是添加一个新的损失函数，同时以 sop_t 为条件； sop_t 是一个取值来自 13 个操作符的潜变量。这可以想象成一个期望最大化（EM）模型：在期望步骤中，计算本文提出的算子预测模型；在最大化步骤中，学习 CopyNet 模型的参数。训练可以分开进行，也可以端到端进行。因此，应最小化以下负似然。

$$L_{enh} = -\frac{1}{N}\sum_{k=1}^{N}\sum_{t=1}^{T}\log[p(y_t^{(k)}|y_{<t}^{(k)}, X^{(k)}, alsop_{<t}^{(k)})] \tag{5.4}$$

5.2 解析图与评分

可以为 QDMR 构建一个新的图库，用于监督学习。基于这种新形式化方法创建图库后，即可轻松利用图评分方法进行解析。

图8显示，该图并非一棵树，因此应使用一般图来表示。处理这一问题的标准方法是将图线性化并转换为树，这样，在计算出树中各节点的分数后，就可以通过标准 CKY 算法求得最优树。尽管

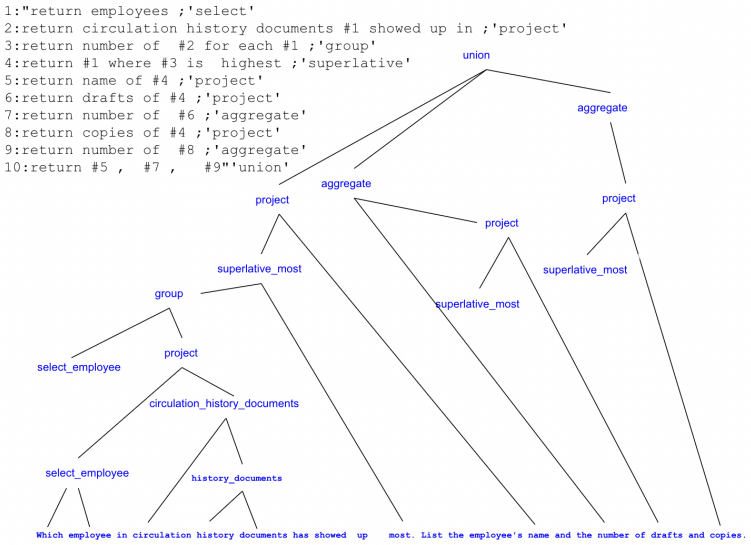

图8：解析图

图8并非一棵树，而是一个有向无环图（DAG），因此无法直接使用 CKY 算法，但可以轻松地使用超图方法（如 Klein & Manning 2001）或基于 Dijkstra算法动态扩展的思路进行解析。

(Socher等人，2010)的方法最初用于句法解析，但经过一些调整后，也可用于语义解析。首先，考虑一个由（句子，树）对组成的数据集，记为 (x_i, y_i)。现在，可以使用组合范畴文法标签（超标签）来代替上下文无关文法（CFG）中的句法标签。每棵树的总分可以计算为各个折叠决策得分之和：

$$s(x_i, y_i) = \sum_{d \in T(y_i)} s_d(c_1, c_2) \tag{5.5}$$

其中， $s_d(c_1, c_2)$ 是树中每个节点的得分

$$\begin{aligned} s_d(c_1, c_2) &= W^{score} p \\ p &= \tanh(W[c_1; c_2]) + b^{(1)}) \end{aligned} \tag{5.6}$$

与分数变量类似，还可以定义另一个标量变量，称为价。节点的价（val）定义为其下方所有文法规则的数量。

$$val_d(c_1, c_2) = W^{val} p \tag{5.7}$$

价的本质是为所有节点建立顺序。终端节点的价为 0，越靠近作为根节点的开始符号，价的数值就越大。因此，根节点的价最高。本文提出的语义算子预测可以表示为一系列价数值；无论树库可用（完全监督），还是树库规模过小（半监督情形），这些数值都是学习过程中的有效特征。应最大化以下目标函数：

$$J = \sum_i s(x_i, y_i) - \max_{y \in A(x_i)} (s(x_i, y) + \Delta(y, y_i)) + val(x_i, y_i) - \max_{y \in A(x_i)} (val(x_i, y) + \Delta(y, y_i)) \tag{5.8}$$

其中， Δ 在树偏离正确树时会对其施加更大的惩罚，其公式如下：

$$\Delta(y, y_i) = \sum_{d \in T(y)} \mathbb{1}\lambda d \notin T(y_i) \tag{5.9}$$

6 结论

本文分析了解决语义解析问题的不同范式，揭示了远程监督的重要性，以及通过创建中间层来避免直接一次性面对问题的方法。这使机器能够分阶段理解句子，从而为语义解析构建更具扩展性的框架。最后，本文提出了一种快速的语义算子预测方法，并展示了该方法在不同模型中的应用。

7 未来工作

提升本文准确率最重要的思路之一，是采用 Abzianidze & Bos 2017 提出的“通用语义标注”，因为本文使用的词性标签无法为词汇语义学提供足够的信息。因此，Abzianidze & Bos 2017 引入了新的类别来解决这一重要问题，包括 13 个元标签和 73 个语义标签。例如，图8中的单词“most”可以用元标签“COM”（表示比较级）和语义标签“TOP”来表示。

参考文献

Abzianidze, L. & Bos, J. (2017), 迈向通用语义标注。

Bahdanau, D., Cho, K. & Bengio, Y. (2014), '通过联合学习对齐与翻译实现神经机器翻译', *CoRR* 。

Berant, J., Chou, A., Frostig, R. &Liang, P.(2013), '基于问答对的 Freebase 语义解析', *EMNLP 2013 - 2013* 自然语言处理实证方法会议论文集第 1533–1544 页。

Dong, L. & Lapata, M. (2018), '神经语义解析的由粗到细解码'。

Gu, J., Lu, Z., Li, H. & Li, V. O. (2016), 在序列到序列学习中引入复制机制，见《第 54 届计算语言学协会年会论文集（第 1 卷：长篇论文）》，计算语言学协会，柏林，德国，第 1631–1640 页。Gu, Y., Kase, S., Vanni, M., Sadler, B., Liang, P., Yan, X. & Su, Y. (2021), 超越独立同分布：知识库问答的三个泛化层次，第 3477–3488 页。Gu, Y., Pahuja, V., Cheng, G. & Su, Y. (2022), '知识库问答：语义解析视角'。Herzig, J. & Berant, J. (2021), 基于跨度的语义解析与组合泛化，第 908–921 页。

Kapanipathi, P., Abdelaziz, I., Ravishankar, S., Roukos, S., Gray, A. G., Astudillo, R. F., Chang, M., Cornelio, C., Dana, S., Fokoue, A., Garg, D., Gliozzo, A., Gurajada, S., Karanam, H. P., Khan, N., Khandelwal, D., suk Lee, Y., Li, Y., Luus, F.P. S., Makondo, N., Mihindukulasooriya, N., Naseem, T., Neelam, S., Popa, L., Reddy, R. R. G., Riegel, R., Rossiello, G., Sharma, U., Bhargav, G. P. S. & Yu, M. (2020), 利用抽象意义表示进行知识库问答，载于"Findings"。

Klein, D. & Manning, C. (2001), '解析与超图',**23**. Krishnamurthy, J., Dasigi, P. & Gardner, M. (2017), 面向半结构化表格的类型约束神经语义解析, pp. 1516–1526. Min, S., Zhong, V., Zettlemoyer, L. & Hajishirzi, H. (2019), 通过问题分解和重新评分实现多跳阅读理解, pp. 6097–6109. Nan, L., Hsieh, C., Mao, Z., Lin, X., Verma, N., Zhang, R., Kry´sci´nski, W., Schoelkopf, N., Kong, R., Tang, X., Mutuma, M., Rosand, B., Trindade, I., Bandaru, R., Cunningham, J., Xiong, C. & Radev, D. (2022), 'Fetaqa：自由形式表格问答', 计算语言学协会会刊**10**, 35–49.

Pasupat, P., Gupta, S., Mandyam, K., Shah, R., Lewis, M. & Zettlemoyer, L. (2019), 面向任务型对话的基于片段的层次语义解析, pp. 1520–1526. Perez, E., Lewis, P., Yih, W.-t., Cho, K. & Kiela, D. (2020), '用于问答的无监督问题分解'. Ravishankar, S., Thai, J., Abdelaziz, I., Mihidukulasooriya, N., Naseem, T., Kapanipathi, P., Rossiello, G. & Fokoue, A. (2021), '一种旨在提升知识库问答泛化能力的两阶段方法'. Ren, H., Dai, H., Dai, B., Chen, X., Yasunaga, M., Sun, H., Schuurmans, D., Leskovec, J. & Zhou, D. (2021), Lego：用于知识图谱多跳问答的潜在执行引导推理, 发表于 M. Meila & T. Zhang, 编, '第38届国际机器学习大会论文集', 第139卷，机器学习研究论文集, PMLR, pp. 8959–8970. Shi, J., Cao, S., Hou, L., Li, J. & Zhang, H. (2021), 'TransferNet：一种用于关系图多跳问答的有效透明框架', *arXiv* 电子预印本p. arXiv:2104.07302. Socher, R., Manning, C. & Ng, A. (2010), '利用递归神经网络学习连续短语表示并进行句法解析'. Sun, H., Bedrax-Weiss, T. & Cohen, W. W. (2019), 'Pullnet：通过在知识库和文本上进行迭代检索实现开放域问答'. Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R. & Cohen, W. W. (2018), 使用知识库和文本早期融合的开放域问答, 发表于 'EMNLP'. Wolfson, T., Geva, M., Gupta, A., Gardner, M., Goldberg, Y., Deutch, D. & Berant, J. (2020), 'Break it down：一个问题理解基准'. Yang, S. & Tu, K. (2022), 使用指针网络进行自底向上成分句法分析和嵌套命名实体识别，发表于 'ACL'. Yavuz, S., Zhou, Y., Keskar, N. & Xiong, C. (2022), '将多跳问答建模为单序列预测'. Zhang, J., Zhang, X., Yu, J., Tang, J., Tang, J., Li, C. & Chen, H. (2022), 用于多跳知识库问答的子图检索增强模型，发表于'第60届计算语言学协会年会论文集（第1卷：长篇论文）', 计算语言学协会 , 都柏林 , 爱尔兰 , pp. 5773–5784. Zheng, H. & Lapata, M. (2020), ' 通过语义标注实现组合泛化'. Zhou, M., Huang, M. & Zhu, X. (2018), 用于多关系问答的可解释推理网络，发表于 '第27届计算语言学国际会议论文集', 计算语言学协会, 圣菲, 新墨西哥州, 美国, pp. 2010–2022.