

# 基于槽依赖建模的跨领域槽填充方法<sup>\*</sup>

王 泽, 周夏冰, 鞠 鑫, 王中卿, 周国栋



(苏州大学 计算机科学与技术学院, 江苏 苏州 215006)

通信作者: 周夏冰, E-mail: [zhouxiabing@suda.edu.cn](mailto:zhouxiabing@suda.edu.cn)

**摘要:** 作为任务型对话系统的一个核心部分, 槽填充任务通过识别话语中存在的特定槽实体来服务于后续的下游任务。但是, 针对一个特定领域, 需要大量有标记的数据作为支撑, 收集成本较高。在此背景下, 跨领域槽填充任务出现, 该任务通过迁移学习的方式高效地解决了数据稀缺问题。已有的跨领域槽填充方法都忽视了槽类型之间在话语中存在的依赖, 导致现有的模型在迁移到新领域时性能欠佳。为了弥补这个缺陷, 提出基于槽依赖建模的跨领域槽填充方法。基于生成式预训练模型的提示学习方法, 设计一种融入槽依赖信息的提示模板, 该模板建立了不同槽类型之间的隐式依赖关系, 充分挖掘预训练模型的实体预测性能。此外, 为了进一步提高槽类型和槽实体与话语文本之间的语义依赖, 增加了话语填充子任务, 通过反向填充的方式增强话语与槽实体的内在联系。通过对多个领域的迁移实验表明, 所提模型在零样本和少样本的设置上取得了较大的性能提升。此外, 对模型中的主要结构进行了详细地分析和消融实验。

**关键词:** 槽填充; 对话系统; 提示学习

中图法分类号: TP18

中文引用格式: 王泽, 周夏冰, 鞠鑫, 王中卿, 周国栋. 基于槽依赖建模的跨领域槽填充方法. 软件学报, 2025, 36(4): 1557–1569.  
<http://www.jos.org.cn/1000-9825/7189.htm>

英文引用格式: Wang Z, Zhou XB, Ju X, Wang ZQ, Zhou GD. Slot Dependency Modeling for Cross-domain Slot Filling. Ruan Jian Xue Bao/Journal of Software, 2025, 36(4): 1557–1569 (in Chinese). <http://www.jos.org.cn/1000-9825/7189.htm>

## Slot Dependency Modeling for Cross-domain Slot Filling

WANG Ze, ZHOU Xia-Bing, JU Xin, WANG Zhong-Qing, ZHOU Guo-Dong

(School of Computer Science and Technology, Soochow University, Suzhou 215006, China)

**Abstract:** This study considers slot filling as a crucial component of task-oriented dialogue systems, which serves downstream tasks by identifying specific slot entities in utterances. However, in a specific domain, it necessitates a large amount of labeled data, which is costly to collect. In this context, cross-domain slot filling emerges and efficiently addresses the issue of data scarcity through transfer learning. However, existing methods overlook the dependencies between slot types in utterances, leading to the suboptimal performance of existing models when transferring to new domains. To address this issue, a cross-domain slot filling method based on slot dependency modeling is proposed in this study. Leveraging the prompt learning approach based on generative pre-trained models, a prompt template integrating slot dependency information is designed, establishing implicit dependency relationships between different slot types and fully exploiting the predictive performance of slot entities in the pre-trained model. Furthermore, to enhance the semantic dependencies between slot types, slot entities, and utterance texts, discourse filling subtask is introduced in this study to strengthen the inherent connections between utterances and slot entities through reverse filling. Transfer experiments across multiple domains demonstrate significant performance improvements in zero-shot and few-shot settings achieved by the proposed model. Additionally, a detailed analysis of the main structures in the model and ablation experiments are conducted in this study to further validate the necessity of each part of the model.

**Key words:** slot filling; dialogue system; prompt learning

\* 基金项目: 国家自然科学基金 (62176174, 61936010)

收稿时间: 2023-11-09; 修改时间: 2024-01-08, 2024-03-13; 采用时间: 2024-03-22; jos 在线出版时间: 2024-06-14

CNKI 网络首发时间: 2024-06-21

任务型对话系统主要分为 4 个部分：自然语言理解、对话状态跟踪、对话策略、对话生成。其中槽填充任务是自然语言理解模块的关键部分，该任务通过抽取用户话语中存在的槽实体为对话系统下游任务提供必要的语义信息。在现实的人机交互应用中，模型对自然语言的理解尤其重要。例如，如果存在话语“Book a tea place for seven”，那么槽填充任务需要识别出话语中的“tea place”和“seven”，并对应到它们在当前领域所属的槽类型，`restaurant_type` 和 `party_size_number`。一些传统的有监督学习方法已经取得了很高的性能<sup>[1-8]</sup>，但是这些方法依赖于大量的高质量标注语料，这个步骤是耗时耗力的。尤其针对一个新兴领域而言，无法在短时间内搜集到大规模的训练数据，难以迁移到新的任务中。因此，跨领域槽填充成为一个必要且极具挑战性的任务，它有效地弥补了数据缺失带来的领域迁移问题<sup>[9]</sup>。

跨领域槽填充任务利用原有领域的数据作为训练集，通过从源域中学习任务知识然后将其迁移到目标领域中进行应用，这个过程不需要目标领域的监督数据作为支撑。如图 1 所示，两个领域中包含若干不同的槽类型（例如 BookRestaurant 领域中的 `restaurant_type`），模型在 BookRestaurant 领域中学习先验知识，然后应用到 PlayMusic 中对新的槽实体进行预测。

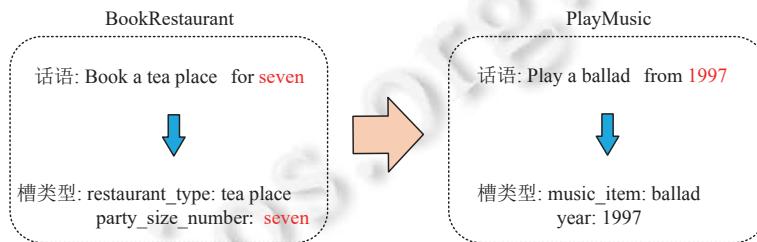


图 1 槽填充任务领域迁移示例

早期的一些研究将槽填充任务形式化为序列标注任务，如图 2(a) 所示，该框架为领域中的每个槽类型分配两个 BI 位置标签<sup>[10,11]</sup>。例如，槽类型 `restaurant_type` 对应有 `B-restaurant_type`, `I-restaurant_type` 两个标签。但是这种方法存在严重的缺陷，随着槽类型数量的增长，标签数量也会成倍上升，对于槽类型数量较多的新领域来说，这种缺陷对性能影响较大。针对这个问题，Liu 等人<sup>[12]</sup>提出了一种两阶段的框架，该方法将实体位置识别和类型匹配分开进行处理，基本解决这一缺陷。后续很多研究也是基于此框架开展的<sup>[13-16]</sup>。考虑到序列标注任务的有限性，一些研究者从一个新的视角出发，将槽填充任务转换为机器阅读理解任务 (MRC)<sup>[17,18]</sup>，如图 2(b) 所示，这种框架为每个槽类型构造一个查询问题，然后与话语文本拼接作为输入，通过预测实体跨度的起始位置来完成槽实体的预测任务。但是，受此任务形式的限制，该方法需要为每个槽类型手工设置查询问题，鲁棒性较差，而且需要使用大量 MRC 数据对预训练模型做二次训练，这样训练数据的选择也会直接影响模型的性能。随着提示学习范式 (prompt learning) 的出现<sup>[19]</sup>，越来越多的任务被转换成生成式问题。Yan 等人<sup>[20]</sup>首次将槽填充任务视为生成任务，他们为每个槽类型构造融合领域和样例信息的槽提示，用于生成相应的槽实体。该框架试图利用提示来激发预训练模型的潜在能力，如图 2(c) 所示。

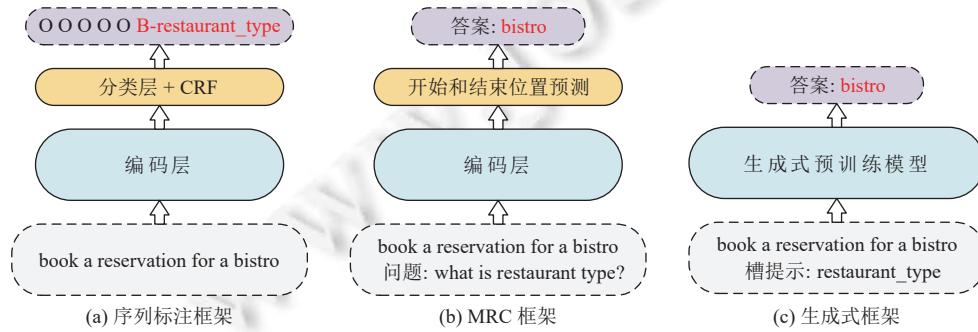


图 2 槽填充任务框架

然而, 前面提到的这些方法都是单独地对槽类型进行预测, 忽略了不同槽类型之间存在的内在联系, 尤其在跨领域场景, 源领域与目标领域之间对槽类型的预测可能存在一些隐式依赖。例如, 在预测天气任务(领域)中, 天气情况(condition\_description)与地点(city)同步出现在用户话语中的可能性是很大的, 这就意味着地点这一槽类型能够辅助模型判断出话语中存在天气情况, 模型在拥有这种意识以后, 对于目标领域(以预定饭店作为目标领域)中的地点(地点属于与源域共享的槽类型)仍然会辅助模型预测与其存在依赖的未知槽类型。比如目标领域的话语中如果提到地点, 那么大概率会同时出现餐厅类型或者餐厅名称等实体, 模型可以借助地点这一槽类型进一步推理。

基于上述分析, 本文提出了一种基于槽依赖建模的跨领域槽填充方法, 该方法基于生成式预训练模型的优势, 首先提出一个新颖的提示模板, 将之前的槽类型与槽实体一对一单独生成的方式改进为多对多生成。该提示模板由槽语义提示和槽共享提示组成, 其中槽语义提示指出要预测的槽类型, 槽共享提示建立槽间依赖, 从而可以捕获不同槽类型之间的潜在关联, 避免独立预测的信息孤立性。其次, 该方法额外增加了一个话语填充子任务, 将槽实体加入到提示中对原文进行填充, 从而增强不同槽类型的实体对原文的感知, 避免出现实体类型匹配错误的问题。

综上, 本文的贡献主要包括以下3点。

(1) 提出了一种基于槽依赖建模的生成式提示学习方法, 利用不同槽类型之间的隐式依赖帮助模型预测相关实体。

(2) 考虑到多个槽类型同时进行实体生成容易出现实体类型错误映射的问题, 本文增加了一个话语填充子任务, 由模型自己选择提示中的实体对原文进行还原。

(3) 在SNIPS和TOP数据集上的实验结果表明本文提出的方法取得了优异的性能, 并且通过多个对比实验验证了各个模块发挥的功能及其存在的必要性。

本文第1节介绍跨领域槽填充任务的相关工作。第2节对生成式跨领域槽填充任务的定义进行阐述, 并且描述基于槽依赖建模的跨领域槽填充方法的实现细节。第3节通过各种实验结果和实验分析表明本文方法的有效性。第4节从整体总结当前方法, 并对未来可改进的部分进行初步探讨。

## 1 相关工作

跨领域槽填充任务主要包括序列标注框架、MRC框架和生成式框架这3种形式。其中, 针对序列标注框架的研究较为充分。跨领域任务最难的一个挑战就是对未知槽类型的预测, 早期的研究中, Ferreira等人<sup>[7]</sup>提出一种在线自适应策略, 使用大量未标注的非结构化数据学习字的连续空间向量表示, 基于神经网络框架将目标域的本体描述和通用词嵌入特征推导出解码模型, 这种方法可以在少量可调整的监督下完善模型。后来研究者将关注点转移到槽类型的描述信息, Bapna等人<sup>[10]</sup>对槽描述信息进行嵌入, 将其融入至多任务框架中训练槽序列标注模型。但是这种方法单独地预测每个槽类型, 容易出现一个实体被预测为多个槽类型的多预测问题。Liu等人<sup>[12]</sup>使用了一种由粗到细的两阶段框架: 第1阶段通过序列标注任务来识别句中存在的槽实体位置, 第2阶段对识别到的槽实体表征与槽类型描述嵌入进行相似度匹配, 此外, 还增加了模板正规化去提升对相似槽类型预测的性能。该方法不仅解决了多预测问题, 而且模型为源域中所有槽类型共享参数, 保证槽实体能学习到通用样式。自两阶段框架出现后, 所有的序列标注任务都开始延用这种结构, 并在此基础上进行改进。He等人<sup>[13]</sup>提出了一种使用对抗攻击的对比零样本学习方法, 通过增加噪声输入来增强跨领域模型的鲁棒性。Liu等人<sup>[14]</sup>分别在两个阶段学习不同的上下文表示, 在槽类型匹配的输入层将槽实体信息融合全局的上下文信息来进行相似度计算。不同领域中存在的公共句法结构也是实现领域迁移的一个关键信息, Liu等人<sup>[15]</sup>将句法树信息嵌入到文本表示中, 在文本表示中融入源域的句法特征, 借助结构关系预测槽实体位置。上述方法都是从增强模型泛化能力的角度去优化。Li等人<sup>[16]</sup>应用原型的概念, 将每个槽描述映射为槽原型, 通过原型对比学习拉近槽实体与其槽原型在语义空间的距离, 提高了类型匹配的准确率。特别地, Luo等人<sup>[21]</sup>在最新研究中, 使用BIO标签作为可训练的序列加入到输入中, 并且将当前领域中的所有槽类型拼接在话语后, 从预训练模型中逐层提取注意力得分进行整合, 作为最终的标注特征。但序列

标注框架仍然存在一定的局限性,该框架需要大量高质量的数据作为支撑,很难迁移到包含较多非共享槽类型的新领域。

随着基于问答的任务不断取得突破,一些研究者从一种全新的视觉出发,将槽填充任务转换为MRC框架。Du等人<sup>[17]</sup>基于槽描述信息为每个槽类型生成问题,并且采用弱监督的方式构造问答数据对BERT进行二次预训练,然后再用于槽填充任务。类似地,Yu等人<sup>[18]</sup>首先将预训练模型在一个大规模的机器阅读理解语料SQuAD2.0中进行预训练,同样基于槽类型描述词和数据样例为每个槽类型设计相应的查询模板,通过训练好的模型对查询中的对应的槽实体位置进行预测。借助大量语料和模板的作用,该方法的性能取得了很大的突破。但是这种框架需要人工构造启发式的模板,模型鲁棒性较差,并且需要借助QA数据对模型进行预热,有很高的时间成本且收集数据代价较大。

近年来,随着提示学习范式的兴起,越来越多的研究者将研究方向转为提示模板的构造。Luo等人<sup>[21]</sup>在输入序列中加入的槽前缀提示也是基于这种思想,同时还分别考虑了冻结预训练模型只训练提示序列和微调全部参数两者的区别,实验结果也证实了提示学习的高效性。Yan等人<sup>[20]</sup>融合了领域知识、槽类型指示信息和若干数据样例作为提示模板,利用生成式预训练模型的潜在能力预测实体。这种融合多种信息的输入模板同样受人工影响较大,因此模型的鲁棒性较差。当新领域信息较为复杂时,输入长度会增加,影响模型性能。并且这种单生成的方式忽略了槽类型之间的潜在依赖信息。

基于上述存在的问题,本文应用提示学习的思想,将单实体生成的方式转换为多实体生成,在提示模板中考虑了不同槽类型之间的隐式依赖关系。进一步通过话语填充任务来减轻多个实体生成过程中存在的槽类型错误匹配问题。

## 2 方法实现

本文所提方法主要基于生成式槽填充方法,下面对相关概念和基本知识以及本文提出的方法进行介绍。

### 2.1 问题定义

对于某个领域 $d(d \in D)$ ,存在一个槽类型集合 $S = [s_1, s_2, \dots, s_n]$ ,槽填充任务的目标是预测话语 $U = [u_1, u_2, \dots, u_n]$ 中存在的所有槽值组合(槽类型和对应的槽实体)。其中,对于话语中不存在的槽类型通常规定模型生成None表示。

### 2.2 生成式槽填充方法

目前生成式预训练模型在各大任务上取得了优异表现,本文的模型也基于此框架进行拓展。传统的生成式框架只是将指定槽类型的提示与话语进行拼接,通过生成式预训练模型(例如:T5、BART)去得到相应的槽实体。对于一个槽类型需要为其选择相应的槽提示文本,如图3(a)所示,可以直接使用槽类型描述作为提示,也可以为每个槽类型分配一个问题文本(例如:restaurant\_name -> what is the restaurant name?)。与传统的生成方式不同,本文采用多实体同步生成的方式,为话语拼接多个槽提示作为输入,如图3(b)所示。

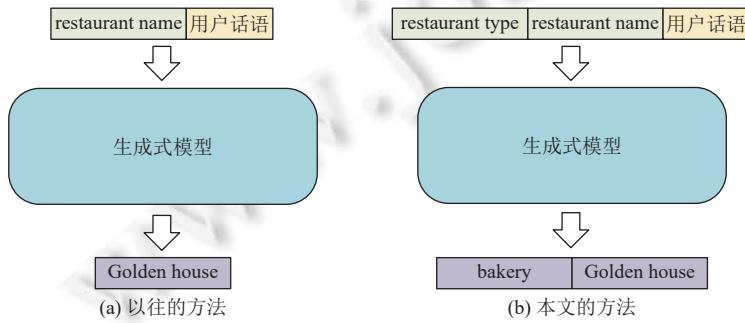


图3 生成式槽填充任务

本文的模型结构如图 4 所示, 主要包括槽填充任务和话语填充子任务两部分。在流程上, 两个任务都是先对槽提示进行重组, 然后输入到生成式模型中得到相应的槽实体序列。输入可以有多个槽提示进行组合, 通过提示整合模块的槽共享提示来建模不同槽类型之间的依赖关系。此外, 本文通过构建话语填充子任务来增强话语中槽实体与槽类型之间的映射关系, 同时进一步强化槽实体与用户话语之间的语义联系。

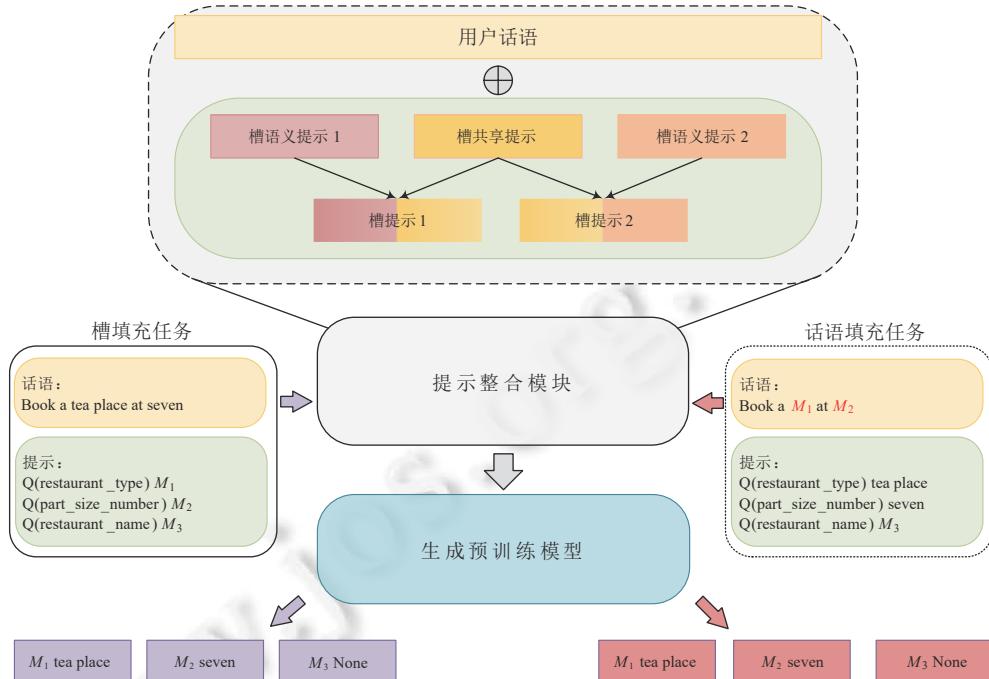


图 4 模型总体框架图

### 2.3 槽依赖建模

为了建模不同槽类型之间的隐式依赖, 本文对多个槽提示进行拼接作为最终的输入提示  $T$ , 每个槽提示都是针对某一槽类型的序列表示。通过一个生成式预训练模型将提示序列  $T$  与第  $t$  个用户话语  $u_t$  进行拼接作为模型输入, 由模型输出  $T$  中相关槽类型所对应的槽实体序列:

$$T = Q(s_1), M_1, \dots, Q(s_n), M_n \quad (1)$$

$$V = Seq2seq(u_t, T) = M_1, v_1, \dots, M_n, v_n \quad (2)$$

公式 (1) 中  $Q(s_i)$  表示将第  $i$  个槽类型  $s_i$  转换为槽提示表示, 具体的转换流程在后面的槽提示模板中进行介绍。模型的输出序列  $V$  中包含所有  $s_i$  所对应的槽实体  $v_i$  ( $i = 1, 2, \dots, n$ ),  $M_i$  表示掩码词。为了帮助模型对每个槽类型进行准确地区分, 我们在输入序列中的每个槽提示后插入  $M_i$ , 对不同槽类型进行分割。同时, 在  $V$  中生成的槽实体前加入  $M_i$ , 使得标识词与输入序列中的槽类型构成映射。通过这种设置能够帮助模型在生成过程中关注到槽类型相对应的槽实体, 直接完成类型的匹配工作。如图 4 中的槽填充任务所示, 用户话语和 3 个槽类型组成的提示模板 (Q(restaurant\_type)  $M_1$  Q(part\_size\_number)  $M_2$  Q(restaurant\_name)  $M_3$ ) 进行拼接后, 生成槽实体序列“ $M_1$  tea place  $M_2$  seven  $M_3$  None”, 其中“tea place”属于槽类型“restaurant\_type”所对应的实体, “seven”属于槽类型“part\_size\_number”所对应的实体, “restaurant\_name”没有对应的实体。

槽实体数量的选择我们在后续进行了实验分析, 在训练阶段通过随机组合槽类型的方式来构建输入, 本文对槽类型进行组合的时候考虑到可能存在的噪声, 比如在文本中本不具备明显依赖关系的两个槽类型强行联合的话会影响性能。因此, 我们基于当前文本实际存在的槽类型进行随机组合, 即对于 None 值也就是不存在的槽类型不需要进行组合。从文本实际存在的槽类型出发, 考虑可能存在的槽依赖, 这样很大程度上降低了多槽融合的噪声问题。

生成式槽填充任务最终的训练目标是最小化下面的损失函数:

$$L = \sum_t \log p(V|u_t, T) \quad (3)$$

#### 2.4 话语填充任务

为了能够更好地建立槽类型与话语中槽实体之间的映射,并且加强槽实体与话语文本之间的潜在依赖关系,本文设计了话语填充子任务。如图 4 所示,该任务同样采用生成的方式利用提示中的实体对话语进行修复生成。本文首先对用户话语中存在的槽实体“tea place”和“seven”进行掩码,将这两个槽实体拼接到其对应的槽提示后,模型通过给定的提示信息对话语中的掩码部分进行填充,话语填充任务的提示表示如下:

$$T' = Q(s_1), v_1, \dots, Q(s_n), v_n \quad (4)$$

$$u'_t = MASK(u_t) \quad (5)$$

$$V' = Seq2seq(u'_t, T') = M_1, v_1, \dots, M_n, v_n \quad (6)$$

具体而言,对于话语中的槽实体使用  $M_i$  进行掩码,依次将  $M_i$  替换到每个实体的位置,然后将第  $i$  个槽实体  $v_i$  拼接到相应的槽提示后,构成该任务的提示模板  $T'$ 。模型将经过掩码后的第  $t$  个用户话语  $u'_t$  与  $T'$  拼接作为模型的输入,模型可以根据提示模板选择相应的槽实体对原文进行填充。同样地,模型生成相应的槽实体序列  $V'$ ,生成序列中使用  $M_i$  作为标识词与话语中的掩码词相呼应,每个  $M_i$  后的实体词是对原文中的掩码部分  $M_i$  的填充词。

此外,对于话语中不存在槽实体的槽类型,在槽类型提示后拼接 None 标识,并且不进行掩码操作,在生成阶段对应的槽实体生成 None 即可。话语填充任务的训练目标是最小化下面的损失函数:

$$L' = \sum_t \log p(V'|u'_t, T') \quad (7)$$

#### 2.5 提示整合模块

对于多个槽提示,为了加强不同槽类型之间的依赖关系,本文提出的方法设计了整合模块来连接多个槽类型。该模块由两部分组成,包括描述槽类型的槽语义提示和槽共享提示。

早期的工作从槽类型的多方信息考虑,包括领域描述信息、针对槽类型本身构造的问题以及槽类型对应的数据示例,这使得建模的过程过于冗余并且有较强的领域依赖性,不够灵活。本文只借助槽类型描述词作为语义提示,帮助模型更好地理解不同的槽类型含义,模型感知一部分先验知识后,进而激发预训练模型的语言理解能力,从而增强模型的槽实体生成性能。表 1 是不同领域槽类型描述。

表 1 不同领域槽类型描述

| 领域             | 槽类型                    | 槽类型描述               |
|----------------|------------------------|---------------------|
| GetWeather     | timeRange              | time range          |
|                | current_location       | current location    |
|                | geographic_poi         | geographic position |
|                | condition_temperature  | temperature         |
|                | condition_description  | weather             |
| BookRestaurant | restaurant_name        | restaurant name     |
|                | restaurant_type        | restaurant type     |
|                | part_size_number       | number              |
|                | served_dish            | served dish         |
|                | party_size_description | person              |

此外,考虑到简单地利用槽类型描述不足以充分构建槽类型之间的联系,本文在整合模块部分增加了可学习的序列作为槽共享提示,从而更好地建立不同槽类型之间的依赖关系。传统的离散提示虽然便于直观的理解(直接使用可读性内容作为提示词),但其构造过程过于主观,并且最终确定的模板不一定是最优效果,仍然可以进行改进。此外,不同槽类型之间存在的隐式依赖很难通过直观的方式人为定义。因此,本文选择使用可学习的嵌入序列来建模不同槽类型共享的提示序列,该序列在训练过程中可以不断进行调整,通过训练逐步接近最优的序列表征。

综上, 本文将两个部分的提示进行拼接作为最终的槽提示模板, 如图 4 所示, 最终每个槽类型的提示序列如公式(8)所示:

$$Q(s_k) = [P_1^k, \dots, P_m^k, P_1^s, \dots, P_l^s] \quad (8)$$

其中,  $[P_1^k, \dots, P_m^k]$  表示第  $k$  个槽类型的槽语义提示序列,  $[P_1^s, \dots, P_l^s]$  表示槽共享提示序列,  $m$  和  $l$  分别表示两者的序列长度,  $m$  的大小由槽类型的语义描述词决定,  $l$  是在实验中预先定义的超参数。所有槽类型都有相同的槽共享提示序列。

为了方便理解, 这里以 BookRestaurant 领域中的槽类型“restaurant\_type”为例, 如果要对其进行预测, 则槽提示模板的文本表示为“restaurant type [  $P^s$  ] [  $P^s$  ]”(这里的槽共享提示长度  $l$  设置为 2), 然后对槽提示进行嵌入, 其嵌入表示为公式(9), 其中  $e_i^k$  ( $i = 1, 2, \dots, m$ ) 是槽语义提示经过预训练模型嵌入层之后得到的嵌入表示,  $h_j^s$  ( $j = 1, 2, \dots, l$ ) 是槽共享提示通过全连接层得到的可学习嵌入张量。

$$E = [e_1^k, \dots, e_m^k, h_1^s, \dots, h_l^s] \quad (9)$$

## 2.6 训练和推理

在训练阶段的损失函数如公式(10)所示, 其中  $\alpha$  是用来控制话语填充子任务所占权重, 本文经过测试最终确定 0.3 为该子任务的最佳权重。

$$L_{\text{train}} = L + \alpha L' \quad (10)$$

在推理阶段, 考虑到目标领域中槽类型数量过于庞大的问题, 如果使用源域中随机组合不同槽类型的方式来构造输入可能会导致组合后的数量难以控制。所以本文仍然使用单个的槽提示与话语拼接对槽实体进行预测。

## 3 实验结果与分析

### 3.1 实验设置

本文使用 T5-base (参数量为 220M) 作为预训练基座模型, batch\_size 的大小设置为 32, 优化器使用 AdamW, 学习率为 1E-4, 子任务权重  $\alpha$  设置为 0.2。对于槽提示的数量  $n$ , 在实际训练过程中我们随机选择  $1 - n$  个槽类型提示用于拼接输入模板, 并且在随机选择过程中保证当前领域中的所有槽类型都能被选择。我们通过实验对不同的槽提示数量进行了测试, 最终确定将  $n$  设置为 2 有最佳性能。同样地, 对于槽共享提示的长度 1 我们也进行了实验测试, 最终将其长度设置为 2。

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \quad (11)$$

在模型训练过程中, 我们使用验证集的  $F1$  值作为监督信号, 如公式(11), 如果超过 3 次模型没有性能提升就结束训练。

### 3.2 基准模型

本文与近几年的一些相关研究进行比较, 这些基线模型的相关情况如下。

Concept tagger (CT)<sup>[10]</sup>: 利用槽描述来处理目标领域中存在的源域不可见的槽类型, 将话语表征与槽类型表征拼接在一起进行预测。

Robust zero-shot tagger (RZT)<sup>[11]</sup>: 在 CT 的基础上引入了槽描述信息和数据样例, 以此来增强零样本槽填充任务的鲁棒性。

Coarse-to-fine approach (COACH)<sup>[12]</sup>: 提出了一种 coarse-to-fine 槽实体检测和槽类型预测方法, 并且使用了模板正规化去提高模型对相似槽类型的性能。

Contrastive zero-shot learning (CZSL)<sup>[13]</sup>: 该方法使用额外的槽描述信息进行对比表征学习, 在第一阶段学习对应槽实体的语义模式, 同时加入对抗攻击的策略增强模型的鲁棒性, 从而提高模型的跨域能力。

Prototypical contrastive learning and label confusion (PCLC)<sup>[16]</sup>: 提出了一种动态细化槽原型表征的方式, 提高槽

实体在类型匹配阶段的性能。

QA-driven slot filling framework (QASF)<sup>[17]</sup>: 将槽填充任务转换为答案抽取任务，并加入了弱监督学习。

Reading comprehension for slot filling (RCSF)<sup>[18]</sup>: 使用大量的 MRC 语料对模型进行预训练，将每个槽类型转换为查询问题，使用模型抽取相应的槽实体。

Abundant information slot filling generator (AISFG)<sup>[20]</sup>: 将槽填充任务转换为生成式任务，使用领域、槽类型、样例信息构造一个新奇的查询模板，引导模型生成相应的槽实体。

Slot-prefix prompting and attention relationship descriptor (SPFT)<sup>[21]</sup>: 在输入中加入槽前缀序列并且使用自定义的注意力描述特征对当前词进行标签预测。

### 3.3 数据集

本文首先在用于槽填充任务主流评测的公开数据集 SNIPS 上进行实验<sup>[22]</sup>。SNIPS 数据集的信息如表 2 所示，该数据集包含 7 个领域，每个领域都有其特定的槽类型，领域之间也存在相同的槽类型，共 39 个。每个领域的训练样本数在 2 000 左右。与以往的研究相同，我们选取其中一个领域的数据作为目标数据，然后选择其他领域的数据作为训练数据，以此来设置领域迁移的场景。数据集中的槽类型分为两个部分，对于不同领域之间共存的槽类型称为领域共享槽类型，对于仅存在于当前领域的槽类型称为领域特定槽类型。为了进一步验证模型的泛化能力和实用性，本文在另一个面向导航和事件查询的语义解析数据集 TOP 上进行验证<sup>[23]</sup>，TOP 中包含 36 个槽类型，其测试集样本数有 9 042 条，复杂度相对较高，更能展现模型的泛化能力。与之前的工作一样，本文使用 SNIPS 中的 7 个领域作为源域数据对 TOP 测试集进行评估。

表 2 SNIPS 数据集的槽类型分布

| 领域                   | 领域共享槽类型   | 领域特定槽类型  |
|----------------------|---|--|
| AddToPlaylist        | music_item, playlist, artist                            | playlist_owner, entity_name  |
| BookRestaurant       | city, state, timeRange, country, sort, spatial_relation | party_size_number, served_dish, restaurant_type, party_size_description, poi, facility, restaurant_name, cuisine |
| GetWeather           | city, state, timeRange, country, spatial_relation       | current_location, geographic_poi, condition_temperature, condition_description                                   |
| PlayMusic            | music_item, playlist, sort, artist                      | genre, service, year, album, track   |
| RateBook             | object_name, object_type                                | object_part_of_series_type, rating_value, object_select, best_rating, rating_unit                                |
| SearchCreativeWork   | object_name, object_type                                | —  |
| SearchScreeningEvent | timeRange, object_type, spatial_relation                | Object_location_type, location_name, movie_name, movie_type  |

### 3.4 实验结果与分析

#### 3.4.1 基线模型比较

表 3–表 5 是本文针对 SNIPS 数据集上各种实验设置的分析。表 3 是在零样本设置下 (zero-shot, 未使用目标领域数据) 将不同领域作为目标领域得到的模型性能的 F1 值。所有基线模型的实验结果都是直接引用原论文的结果。对于 SPPT 方法，原文中有两种实现方式，我们引用其中的一种方法。可以看到，本文提出的方法已经超过了所有的基线模型。

相比于传统的序列标注任务，本文提出的方法已经超过了其中最好的模型 SPPT 将近 10%，并且 SPPT 也使用了预训练模型作为基座模型，由此可见，通过槽提示生成模型更能精确地抽取话语中存在的槽实体，预训练模型得到的先验知识比序列标注的方式更充分且更有针对性。同时，解决了标签爆炸的问题。与 MRC 任务中最好的模型 RCSF 相比，本文的方法高出将近 2%，只有两个领域效果略差。也可以从侧面反映出提示模板更能挖掘预训练模型中的潜在的知识。此外，本文的方法不需要在大批量的 MRC 数据集上进行额外的预训练操作，更加简单高效。与同是生成模型的 AISFG 任务相比，本文的方法超过其性能 3% 左右，并且几乎在所有领域上的效果都要优于

AISFG. 我们复现了对话状态跟踪的一个生成模型 T5DST<sup>[24]</sup>, 该方法考虑了不同形式的槽类型描述作为 prompt. 实验结果足以证明本文策略的有效性.

表 3 SNIPS 数据集中不同模型的 F1 (%)

| 领域   | 序列标注任务 |       |       |       |       | MRC任务 |       |              | 生成任务   |              |              |
|------|--------|-------|-------|-------|-------|-------|-------|--------------|--------|--------------|--------------|
|      | CT     | RZT   | Coach | CZSL  | PCLC  | SPPT  | QASF  | RCSF         | T5DST* | AISFG        | Ours         |
| ATP  | 38.82  | 42.77 | 50.90 | 53.89 | 59.24 | 58.16 | 59.29 | <b>68.70</b> | 58.05  | 56.20        | 58.36        |
| BR   | 27.54  | 30.68 | 34.01 | 34.06 | 41.36 | 44.94 | 43.13 | 63.49        | 50.18  | 65.94        | <b>67.22</b> |
| GW   | 46.45  | 50.28 | 50.47 | 52.04 | 54.21 | 66.06 | 59.02 | 65.36        | 68.26  | 67.66        | <b>75.35</b> |
| PM   | 32.86  | 33.12 | 32.01 | 34.59 | 34.95 | 36.53 | 33.62 | <b>53.51</b> | 41.23  | 50.12        | 52.28        |
| RB   | 14.54  | 16.43 | 22.06 | 31.53 | 29.31 | 28.02 | 33.34 | 36.51        | 30.10  | <b>41.05</b> | 39.39        |
| SCW  | 39.79  | 44.45 | 46.65 | 50.61 | 53.51 | 67.77 | 59.90 | 69.22        | 66.70  | 67.46        | <b>69.81</b> |
| SSE  | 13.83  | 12.25 | 25.63 | 30.05 | 27.17 | 34.54 | 22.83 | 33.54        | 34.01  | 35.04        | <b>40.85</b> |
| 平均F1 | 30.55  | 32.85 | 37.39 | 40.99 | 42.82 | 48.00 | 44.45 | 55.76        | 49.79  | 54.78        | <b>57.61</b> |

注: \*表示复现的结果

表 4 SNIPS 数据集中不同模型在零样本设置下 seen 和 unseen 的平均 F1 (%)

| 测试集    | CT    | RZT   | Coach | PCLC  | QASF  | RCSF         | Ours         |
|--------|-------|-------|-------|-------|-------|--------------|--------------|
| seen   | 37.23 | 40.99 | 46.22 | 51.69 | 56.23 | <b>75.96</b> | 74.22        |
| unseen | 3.38  | 2.19  | 9.31  | 17.38 | 41.73 | 26.21        | <b>47.37</b> |

表 5 SNIPS 数据集中不同模型在 50 少样本设置下的平均 F1 (%)

| 模型   | CT    | RZT   | Coach | PCLC  | AISFG | Ours         |
|------|-------|-------|-------|-------|-------|--------------|
| 平均F1 | 64.85 | 66.67 | 75.51 | 78.62 | 84.39 | <b>88.90</b> |

对于其中提升较多的 GetWeather, 其槽类型大多是共享槽类型可迁移性较好, 而且领域特定槽类型像地理位置 (geographic\_poi)、温度 (condition\_temperature)、当前位置 (current\_location) 等, 虽然槽类型命名不同但都是常见类型, 因此相关实体的知识很容易从源域或者预训练模型中得到, 只需正确与其他类型进行区分即可. RateBook 效果最差, 该领域较为独立并且槽类型与其他领域联系较少, 很难建立联系.

下面针对其他性能做出更加细致地分析.

根据之前相关研究中的定义, 在目标领域的测试集中不存在领域特定槽类型的数据叫做可见集 (seen), 存在领域特定槽类型的数据叫不可见集 (unseen). 对于 SearchCreativeWork 领域只存在可见集, 因此只能计算其 seen 结果. RateBook 只存在不可见集, 只计算其 unseen 结果. 如表 4 所示, 可以看到本文的模型在不可见集上取得了相对出色的效果, 原因是本文采用的槽语义信息作为提示有更强大的泛化能力, 槽依赖信息更能引导模型对未知槽的发现. 此外, 我们在可见集上的表现与 RCSF 相差 1.74%, 主要原因是 RCSF 提前在大量的 MRC 任务上进行了二次训练, 对于可见集的把控能力要更强, 而我们的模型没有这一步操作, 但我们的模型仍然比同一任务形式下的 QASF 模型要高出 17.92%.

为了进一步测试模型在少样本设置 (few-shot) 下的效果, 本文使用 50 作为少样本数量, 将目标领域的数据加入到训练数据集中, 对模型的少样本迁移能力进行分析. 如表 5 所示. 与已有的对少样本进行测试的方法相比, 可以看到我们的方法取得了相对明显的性能提升, 超过 AISFG 任务 4.51%. 无论是零样本还是少样本设置, 我们的模型都具有更强的领域迁移能力. 此外, 引入更多的目标域数据能帮助模型准确掌握目标域和源域之间槽依赖存在的联系, 有助于发挥该方法性能.

以上仅是针对同一数据集的分析, 考虑到同一个数据集中虽然领域和槽类型有所不同, 但仍然存在联系. 为了验证模型在更广泛的场景和多样化数据下的效果, 本文针对模型增加了从 SNIPS 到 TOP 的迁移分析. 表 6 是在不同少样本设置下的实验结果. 与同样使用 TOP 进行跨领域的测试的几个方法相比, 可以看到本文的模型在零样本和 50 少样本设置下的都取得了更好的效果, 并且超过了加入弱监督数据的 QASF 方法. 本文的模型使用更小的代价更高效地实现了跨数据集的泛化.

表 6 TOP 数据集中不同少样本设置下不同模型的  $F1$  (%)

| 模型                         | zero-shot    | 50-shot      |
|----------------------------|--------------|--------------|
| ZSBT                       | 8.82         | 42.73        |
| QASF                       | 10.27        | 46.49        |
| QASF w/ pre-training on WS | 12.35        | 47.91        |
| Ours                       | <b>13.55</b> | <b>51.02</b> |

### 3.4.2 不同参数设置

本文采用了多实体生成的方式, 所以对不同领域中的话语存在的槽实体数量进行了统计, 其中对 TOP 数据集我们只统计了其测试集的数量分布, 如表 7 所示。可以看到大部分话语中存在的实体数量在 1–3 个, 可以侧面反映出 SNIPS 和 TOP 这两个任务型数据集的单句文本中槽类型的强依赖组合保持在 3 个以内。为了验证最适合建模槽依赖的实体数量, 本文分别对不同槽数量构成的提示模板进行了实验。如表 8 所示, 槽类型数量设置为 2 时模型出现了最佳效果, 由此推断槽类型之间的隐式依赖更多地出现在两个槽类型之间。设置为 1 时, 也就是之前的槽实体独立生成方式, 缺少了槽类型之间的依赖信息, 模型的性能有较大幅度的下降。相比较而言, 槽类型数量设置为 3 的效果更好, 依然能超过 AISFG 任务。槽依赖信息的加入相比较 AISFG 中的领域和样例等多方信息更能发挥模型的效果, AISFG 只是简单地增加了更多的启发知识, 而且人工构建的模板很难达到最佳性能, 我们通过可训练的槽共享提示序列不仅引入了依赖信息, 而且可以不断调整到最佳表征。

表 7 不同领域共现槽数量统计

| 领域  | 话语中的槽实体数(总数)  |
|-----|---|
| ATP | 2 (754), 3 (1111), 4 (177)                                  |
| BR  | 1 (57), 2 (357), 3 (1069), 4 (320), 5 (178), 6 (80), 7 (12) |
| GW  | 1 (358), 2 (914), 3 (673), 4 (155)                          |
| PM  | 1 (521), 2 (862), 3 (531), 4 (185), 5 (1)                   |
| RB  | 2 (158), 2 (611), 4 (880), 5 (400), 6 (7)                   |
| SCW | 1 (516), 2 (1538)   |
| FSE | 1 (436), 2 (542), 3 (637), 4 (99), 5 (6)                    |
| TOP | 0 (146), 1 (101), 2 (172), 3 (59), 4 (16), 5 (6)            |

表 8 不同槽数量结果

| 槽类型数量 | 平均 $F1$ (%)  |
|-------|--------------|
| 1     | 53.55        |
| 2     | <b>57.61</b> |
| 3     | 55.79        |

槽共享提示使用可学习的嵌入序列, 可以用一个或多个嵌入来实现, 所以研究不同长度的槽共享提示是十分重要的。如图 5 所示, 我们通过改变不同的槽共享提示的长度来对模型的性能进行测试。我们将槽共享提示的长度设为 1 时, 可以看到其性能远低于其他长度。并且经过测试可以看出, 模型的性能不会随着槽共享提示序列长度的增加成正比例上升, 而是呈现出一定的波动。我们选择长度为 2 作为最终的设置。

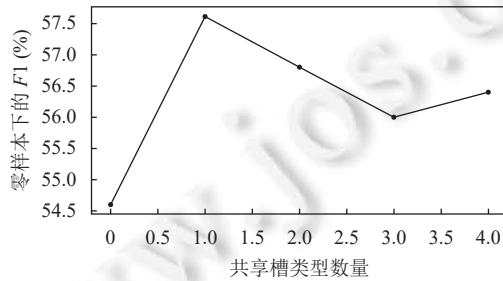


图 5 不同槽共享提示长度

### 3.4.3 消融实验

表 9 是对本文方法的消融实验结果, 失去槽共享提示后的模型只是对多个槽类型同时进行预测, 模型得到的提示知识只包含槽类型的描述信息, 先验知识比较匮乏, 导致模型性能下降了 3.05%。我们又将话语填充子任务去

掉进行测试, 对于多槽类型的生成问题, 需要挖掘不同槽类型与话语中存在的语义感知, 否则难免会出现的错误类型匹配, 受此影响该部分性能损失了 2.12%.

表 9 零样本下的消融实验结果 (%)

| 模型          | 平均F1         |
|-------------|--------------|
| w/o 话语填充子任务 | 55.49        |
| w/o 槽共享提示   | 54.56        |
| Ours        | <b>57.61</b> |

可以得出结论, 使用槽共享提示建立的不同槽类型之间的依赖关系对跨领域槽实体预测起到了很大的作用,直接影响着整个模型的性能. 往往用户在提出一个请求时需要同时指定多个必要条件, 正如下文样例分析中的两个例子, 在预定饭店时 (BookRestaurant 领域) 除饭店类型外具体地点也是必要条件, 在大多数领域中地点都是一个关键信息, 比如 GetWeather、SearchScreeningEvent、TOP 等领域中均有位置信息, 这种依赖很容易从源域中得到. 话语填充子任务的存在也有效地弥补了多实体预测中存在的槽类型错误匹配的问题, 对相似槽类型做出正确判断, 进而影响其他相关依赖槽, 二者缺一不可.

#### 3.4.4 样例分析

如图 6 所示, 本文对模型的生成结果进行了对比, 图中的 baseline 表示去除槽共享提示和话语填充子任务后的基础模型, gold 是真实的数据标签. 对于 BookRestaurant 中的示例, baseline 能够预测出槽类型 city 和 spatial\_relation, 但是缺失了 restaurant\_type, 根据该领域的话语习惯, 通常用户在提供地点或者具体位置后往往会继续给出餐厅的具体类型. 存在 restaurant\_type 的用户话语中有 87% 伴随着 city 或者 spatial\_relation, 因此 3 个槽类型彼此间存在依赖, baseline 模型并没有从另外两个已存在的槽类型中推测出 restaurant\_type, 可见其缺乏这种槽间依赖的信息. 我们的模型很好地解决了这一问题, 侧面反映出槽依赖的必要性.

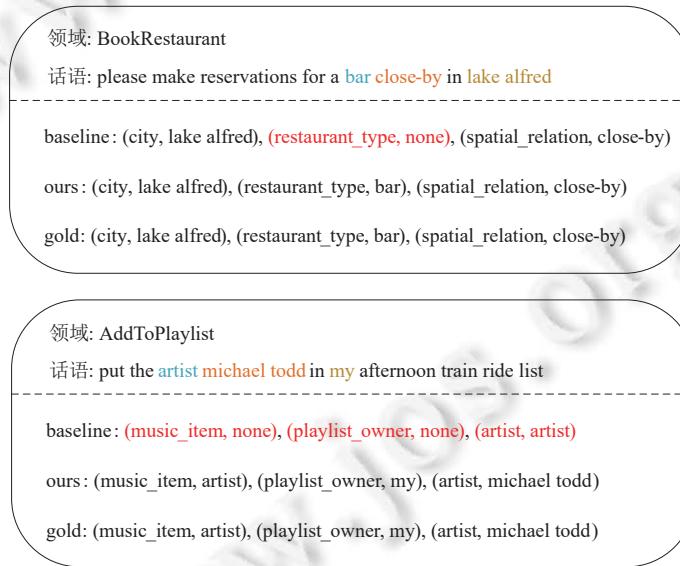


图 6 模型结果样例对比

在 AddToPlaylist 的示例中, baseline 模型出现了错误匹配问题, 它将话语中的实体词“artist”匹配到了槽类型 artist, 虽然两个词一样但是表示的意义不同, 槽类型 artist 对应的槽实体应该是话语中存在的人名, 而实体词 artist 本身应该属于槽类型 music\_item, 这两个类型在预测时容易出现混淆. 我们的模型通过话语填充子任务弥补了这一现象, 在训练过程中使用真实的槽实体对掩码后的话语进行还原, 加强了槽实体对话语的感知, 还原的过程中也

会对不同槽实体的位置进行区分。而且 artist、music\_item 均属于领域共享槽类型，显然模型可以已经从源域中获得这方面的知识。经过统计，当数据中存在 artist 时有 38% 的概率存在 music\_item，有 58% 的概率存在 playlist\_owner。artist 与两者有很强的依赖关系，我们的模型会按照潜在的语言习惯选择合适的答案片段，而不像 baseline 在预测错误后直接影响另一个强相关的槽类型。

## 4 总 结

本文提出了基于槽依赖建模的跨领域槽填充方法，该方法通过生成式的框架同时对多个槽类型进行预测，并且每个槽类型的提示序列包含槽语义提示和槽共享提示两部分，从而引入了槽语义知识和不同槽之间的隐式依赖，不同槽类型之间可以互相辅助生成。对于多实体生成带来的实体类型匹配问题，我们设计了话语填充子任务来弥补这一缺陷。通过填充掩码后的话语，增强了实体与话语的语义感知，进而提升了类型匹配的准确率。在 SNIPS 和 TOP 上的实验结果证实了本文模型在各个方面的性能优势，并且通过各种比较实验和示例分析验证了模型各部分的有效性。在后续的研究中，我们会进一步挖掘任务中存在的其他依赖或潜在信息，尽可能从多个角度为模型提供信息。

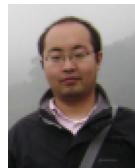
### References:

- [1] Goo CW, Gao G, Hsu YK, Luo CL, Chen TC, Hsu KW, Chen YN. Slot-gated modeling for joint slot filling and intent prediction. In: Proc. of the 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 2 (Short Papers). New Orleans: ACL, 2018. 753–757. [doi: [10.18653/v1/N18-2118](https://doi.org/10.18653/v1/N18-2118)]
- [2] Li X, Wang YY, Tür G. Multi-task learning for spoken language understanding with shared slots. In: Proc. of the 12th Annual Conf. of the Int'l Speech Communication Association. Florence: ISCA, 2011. 701–704.
- [3] Yazdani M, Henderson J. A model of zero-shot learning of spoken language understanding. In: Proc. of the 2015 Conf. on Empirical Methods in Natural Language Processing. Lisbon: ACL, 2015. 244–249. [doi: [10.18653/v1/D15-1027](https://doi.org/10.18653/v1/D15-1027)]
- [4] Zhang CW, Li YL, Du N, Fan W, Yu P. Joint slot filling and intent detection via capsule neural networks. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019. 5259–5267. [doi: [10.18653/v1/P19-1519](https://doi.org/10.18653/v1/P19-1519)]
- [5] Qin LB, Liu TL, Che WX, Kang BB, Zhao SD, Liu T. A co-interactive Transformer for joint slot filling and intent detection. In: Proc. of the 2021 IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP). Toronto: IEEE, 2021. 8193–8197. [doi: [10.1109/ICASSP39728.2021.9414110](https://doi.org/10.1109/ICASSP39728.2021.9414110)]
- [6] Wu D, Ding L, Lu F, Xie J. SlotRefine: A fast non-autoregressive model for joint intent detection and slot filling. In: Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP). ACL, 2020. 1932–1937. [doi: [10.18653/v1/2020.emnlp-main.152](https://doi.org/10.18653/v1/2020.emnlp-main.152)]
- [7] Ferreira E, Jabaian B, Lefèvre F. Online adaptative zero-shot learning spoken language understanding using word-embedding. In: Proc. of the 2015 IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP). South Brisbane: IEEE, 2015. 5321–5325. [doi: [10.1109/ICASSP.2015.7178987](https://doi.org/10.1109/ICASSP.2015.7178987)]
- [8] E HH, Niu PQ, Chen ZF, Song MN. A novel bi-directional interrelated model for joint intent detection and slot filling. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019. 5467–5471. [doi: [10.18653/v1/P19-1544](https://doi.org/10.18653/v1/P19-1544)]
- [9] Li XC, Wang YJ, Gan L, Zhan DC. Exploring transferability measures and domain selection in cross-domain slot filling. In: Proc. of the 2022 IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP 2022). Singapore: IEEE, 2022. 3758–3762. [doi: [10.1109/ICASSP43922.2022.9746890](https://doi.org/10.1109/ICASSP43922.2022.9746890)]
- [10] Bapna A, Tür G, Hakkani-Tür D, Heck L. Towards zero-shot frame semantic parsing for domain scaling. In: Proc. of the 18th Annual Conf. of the Int'l Speech Communication Association. Stockholm: ISCA, 2017. 2476–2480.
- [11] Shah D, Gupta R, Fayazi A, Hakkani-Tür D. Robust zero-shot cross-domain slot filling with example values. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019. 5484–5490. [doi: [10.18653/v1/P19-1547](https://doi.org/10.18653/v1/P19-1547)]
- [12] Liu Z, Winata GI, Xu P, Fung P. Coach: A coarse-to-fine approach for cross-domain slot filling. In: Proc. of the 58th Annual Meeting of the Association for Computational Linguistics. ACL, 2020. 19–25. [doi: [10.18653/v1/2020.acl-main.3](https://doi.org/10.18653/v1/2020.acl-main.3)]
- [13] He KQ, Zhang JC, Yan YM, Yan YM, Xu WR, Niu C, Zhou J. Contrastive zero-shot learning for cross-domain slot filling with adversarial attack. In: Proc. of the 28th Int'l Conf. on Computational Linguistics. Barcelona: ACL, 2020. 1461–1467. [doi: [10.18653/v1/2020.coling-main.126](https://doi.org/10.18653/v1/2020.coling-main.126)]

- [14] Liu SD, Huang PJ, Zhu ZB, Zhang HL, Tan JY. Cross-domain slot filling with distinct slot entity and type prediction. In: Proc. of the 10th CCF Int'l Conf. on Natural Language Processing and Chinese Computing. Qingdao: Springer, 2021. 517–528. [doi: [10.1007/978-3-030-88480-2\\_41](https://doi.org/10.1007/978-3-030-88480-2_41)]
- [15] Liu LC, Lin XX, Zhang P, Wang B. Improving cross-domain slot filling with common syntactic structure. In: Proc. of the 2021 IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP 2021). Toronto: IEEE, 2021. 7638–7642. [doi: [10.1109/ICASSP39728.2021.9414625](https://doi.org/10.1109/ICASSP39728.2021.9414625)]
- [16] Li JN, Zhou P, Xiong CM, Hoi S. Prototypical contrastive learning of unsupervised representations. In: Proc. of the 9th Int'l Conf. on Learning Representations. Vienna: OpenReview.net, 2021.
- [17] Du XY, He LH, Li Q, Yu D, Pasupat P, Zhang Y. QA-driven zero-shot slot filling with weak supervision pretraining. In: Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int'l Joint Conf. on Natural Language Processing (Vol. 2: Short Papers). ACL, 2021. 654–664. [doi: [10.18653/v1/2021.acl-short.83](https://doi.org/10.18653/v1/2021.acl-short.83)]
- [18] Yu MS, Liu J, Chen YF, Xu JA, Zhang YJ. Cross-domain slot filling as machine reading comprehension. In: Proc. of the 30th Int'l Joint Conf. on Artificial Intelligence. Montreal: ijcai.org, 2021. 3992–3998. [doi: [10.24963/ijcai.2021/550](https://doi.org/10.24963/ijcai.2021/550)]
- [19] Ge CJ, Huang R, Xie MX, Lai ZH, Song SJ, Li S, Huang G. Domain adaptation via prompt learning. arXiv:2202.06687, 2022.
- [20] Yan Y, Ye JD, Zhang ZB, Wang LW. AISFG: Abundant information slot filling generator. In: Proc. of the 2022 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Seattle: ACL, 2022. 4180–4187. [doi: [10.18653/v1/2022.naacl-main.308](https://doi.org/10.18653/v1/2022.naacl-main.308)]
- [21] Luo QY, Liu LQ. Zero-shot slot filling with slot-prefix prompting and attention relationship descriptor. In: Proc. of the 37th AAAI Conf. on Artificial Intelligence. Washington: AAAI Press, 2023. 13344–13352. [doi: [10.1609/aaai.v37i11.26566](https://doi.org/10.1609/aaai.v37i11.26566)]
- [22] Coucke A, Saade A, Ball A, Bluche T, Caulier A, Leroy D, Doumouro C, Gisselbrecht T, Caltagirone F, Lavril T, Primet M, Dureau J. Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces. arXiv:1805.10190, 2018.
- [23] Gupta S, Shah R, Mohit M, Kumar A, Lewis M. Semantic parsing for task oriented dialog using hierarchical representations. In: Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing. Brussels: ACL, 2018. 2787–2792. [doi: [10.18653/v1/D18-1300](https://doi.org/10.18653/v1/D18-1300)]
- [24] Lin ZJ, Liu B, Moon S, Crook P, Zhou ZP, Wang ZG, Yu Z, Madotto A, Cho E, Subba R. Leveraging slot descriptions for zero-shot cross-domain dialogue state tracking. In: Proc. of the 2021 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. ACL, 2021. 5640–5648. [doi: [10.18653/v1/2021.naacl-main.448](https://doi.org/10.18653/v1/2021.naacl-main.448)]



王泽(1999—), 男, 硕士生, 主要研究领域为自然语言处理。



王中卿(1987—), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为自然语言处理。



周夏冰(1988—), 女, 博士, 副教授, CCF 专业会员, 主要研究领域为自然语言处理。



周国栋(1967—), 男, 博士, 教授, CCF 杰出会员, 主要研究领域为自然语言处理。



鞠鑫(2001—), 男, 硕士生, 主要研究领域为自然语言处理。