

# 基于缓解错误传递策略的对话状态跟踪

胡峻源  
黑龙江大学  
hujunyuan1169@foxmail.com

周小璐  
哈尔滨工业大学  
xiaoluha2016@163.com

谭龙\*  
黑龙江大学  
tanlong@hlju.edu.cn

## 摘要

对话状态跟踪模块是任务型对话系统的核心组件。现有的一些对话状态跟踪方法基于上一轮的对话状态生成轮级状态，存在错误传递的问题，会对后续预测产生影响。因此本文提出了个基于缓解错误传递策略的对话状态跟踪模型，该模型使用对话级状态作为预测目标，在模型训练时以一定的概率随机删除前一轮次的对话状态，迫使模型在不完全可信的对话状态信息中学会纠正错误。本文在嘈杂(MultiWOZ 2.1) 和洁净(MultiWOZ 2.4) 数据集的实验表明，该模型相比较于基线模型有更好的错误修正能力，模型的联合准确率(MultiWOZ 2.4)达到了70.95%的良好性能表现。

**关键词：** 对话状态跟踪；任务型对话系统；DST

## Dialogue State Tracking Based on Error Propagation Mitigation Strategy

Junyuan Hu  
Heilongjiang University  
hujunyuan1169@foxmail.com

Xiaolu Zhou  
Harbin Institute of Technology  
xiaoluha2016@163.com

Long Tan\*  
Heilongjiang University  
tanlong@hlju.edu.cn

## Abstract

The dialogue state tracking module is a core component of task-oriented dialogue systems. Some existing dialogue state tracking methods generate turn-level states based on the previous dialogue state, which may lead to error propagation and affect subsequent predictions. Therefore, this paper proposes a model based on a strategy to mitigate error propagation. The model's prediction target is changed from turn-level states to dialogue-level states, using less historical dialogue information. During model training, the previous dialogue state is randomly removed with a certain probability, reducing the dependence on historical dialogue states. Experiments on noisy (MultiWOZ 2.1) and clean (MultiWOZ 2.4) datasets show that this method can effectively alleviate error propagation, and the model achieves a good performance with a joint accuracy of 70.95% on the MultiWOZ 2.4 dataset.

**Keywords:** dialogue state tracking, task-oriented dialogue system, DST

©2023 中国计算语言学大会  
根据《Creative Commons Attribution 4.0 International License》许可出版

## 1 引言

任务型对话系统旨在帮助用户完成特定的任务，已在大众日常生活中有广泛应用，例如Siri、Google Assistant、Xiaoice等，受到学术界和工业界的广泛关注。任务型对话系统的构建主要分为两类方法，一种是基于管道(pipeline)的方法，该类方法将任务对话系统分为四个级联模块；另一种是基于端到端的方法，该类方法试图直接通过用户输入产生输出，目前该类方法的精度和健壮性还有待提高(Takanobu et al., 2020)。对话状态跟踪(Dialogue state tracking, DST)是基于管道的任务型对话系统的核心组件，它根据整个对话历史记录跟踪用户的目标和槽值对，以提供策略学习模块决定智能体采取行动的信息(Chen et al., 2017)。由于对话的成功受到系统捕捉用户需求的能力的影响，一个准确的状态跟踪对于任务对话系统是至关重要的(Kim et al., 2018)。简单的对话状态跟踪示例如图1所示。

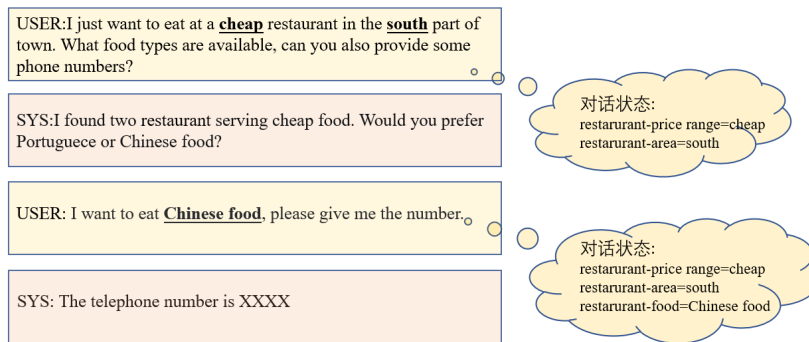


Figure 1: 对话状态跟踪的简单示例

近些年，基于神经网络的DST模型可以分为两大类，第一类是有预定义的槽名和候选槽值列表，每轮DST模块都尝试根据对话选择预定义列表中最合适的槽值对，该类方法也被称为基于本体的DST (ontology-based DST) (Lee et al., 2019; Ye et al., 2021);第二类DST模型则没有固定的槽值列表，因此该类模型试图直接从对话上下文找到值或根据对话上下文直接生成槽值(Kim et al., 2020; Heck et al., 2020),该类方法也被称为基于开放词汇表的DST (open vocabulary-based DST) (Kim et al., 2020)。第一类DST可以看做是一个多跳多分类问题，但由于其需要预定义槽值则带来了许多局限性：(1)在实际应用中，有很多槽位是无法实现获取到完整的槽值列表的，例如无穷举的时间序列和用户口中不在槽值列表中的值等。(2)随着对话数据规模的上升，对话系统多领域对话需求的提高，也会给预定义槽值列表带来更多困难，这不光带来人工的成本增加，也增加了算法的复杂度。(3)由于模型是基于预定义好的槽值训练的，当遇到模型需要重新调整的时候，往往带来很大的局限性。正是因为预定义槽值方法的局限性，开放式槽值表的方法获得了越来越多的研究和关注。这种开放式的方法提供了可伸缩性，并且能够处理不可见的插槽值。

在基于开放词汇表的DST之中，基于数据驱动的神经对话状态追踪算法的主流做法是将整个对话历史作为输入，然后在每轮次从头预测对话状态，然而这种做法往往会带来更高的计算复杂度和数据稀疏性问题(Zhu et al., 2020; Wu et al., 2019)。为了避免以上问题，最近的一类方法将DST分解为两个显式的子任务：槽位操作状态操作预测和槽值生成。这种方法将对话状态视为一个可选择性覆盖的记忆，在每个轮次中，由槽状态操作预测决定是否或如何修改前一个对话状态中的值，并通过槽值生成器进行解码生成槽值(Kim et al., 2020; Zeng and Nie, 2020)。虽然该类方法通过预测轮次级状态(turn-level state)(Balaraman et al., 2021)，来辅助模型降低重头推理的难度，避免了一次性生成全部对话状态的高昂开销，但是在这种马尔科夫假设下，也不可避免的造成了错误的传递。当前轮次如果预测错误，会成为下一轮次的输入，导致模型预测出错误的结果。随着对话轮次的增长，DST追踪成功率也会随之降低。之所以会出现这种现象，本质原因是模型执行的是一次性生成，模型没有机会或很难及时更正错误。

在本文中，我们提出了一种基于缓解错误传递策略的对话状态跟踪模型：DLSD-DST，该模型在面向槽值候选表开放的情景下，将DST任务分为两个显式的子任务，即槽位操作状态预测和槽值生成，模型的预测目标为对话级状态(Balaraman et al., 2021)，在训练的时候以

一定概率删除前一轮次对话状态，并只使用少量对话历史作为辅助输入。实验表明，该模型在MultiWOZ 2.1(Eric et al., 2020) 和MultiWOZ 2.4(Ye et al., 2022) 数据集上都获得了具有竞争力的结果。通过消融实验的验证，模型的性能提升是来自于历史错误的修正。该方法具有普适性，可以应用在很多DST模型中，相较于全历史模型输入规模更少。

## 2 相关工作

近年来，基于神经网络设计的DST模型，特别是预训练的语言模型已成为主流，并提出了大量的神经DST模型。传统的方法将DST任务看成多分类任务，但是由于固定的槽值列表带来很多局限性，很多DST研究开始转向基于开放词汇表的方法(Chen et al., 2017)。最初一些开放槽值列表的模型一开始在每个对话轮次中都需要从头开始多次重复生成每个槽位的对话状态，加之需要较完整的历史对话，这无疑给模型带来了很大的负担(Wu et al., 2019)。

为了解决这个问题，Kim et al. (2020)将DST分为两个子任务，槽位状态操作预测和槽值生成。槽位状态操作预测用于预测该轮次每个槽位应该执行的操作，将不需要生成的槽位进行过滤，避免了重头开始预测。槽值生成任务则根据状态操作预测的结果生成槽值。Lin et al. (2020)将当前轮次的对话和之前的对话状态作为输入序列，将对话状态跟踪作为因果语言模型，利用编码器-解码器框架依次生成轮次级状态和系统响应。Yang et al. (2021)使用每个对话轮次的用户话语、信念状态、数据库结果、系统行为和系统响应组成的整个对话序列作为输入来生成轮次级状态。

为了缓解使用轮次级状态产生带来的错误传递现象，Tian et al. (2021)提出新的可修改对话状态的模型，模型分两步生成：(1) 根据当前轮次的对话和上一轮次的对话状态生成当前轮次的原始对话状态 (2) 修改第一步生成的原始对话状态。模型的任务是通过修正原始对话状态中仍然存在的错误来学习更鲁棒的对话状态跟踪，在双重检查过程中起到修正者的作用，并减少不必要的错误传递。

## 3 模型设计

### 3.1 对话状态跟踪任务形式化描述

对话状态跟踪任务目标是在对话的每个轮次从系统响应和用户话语中提取一组槽值对。这些槽值对的组合形成了一个对话状态，它跟踪用户告知系统的完整意图或需求。

$P = [(s_1, u_1), \dots, (s_t, u_t), \dots, (s_T, u_T)]$ 是长度为 $T$ 的对话序列，其中 $s_t$ 和 $u_t$ 分别代表 $t$ 轮次时的系统响应和用户话语。有一组 $n$ 个已经预定好的域槽对 $S = \{s_1, \dots, s_i, \dots, s_n\}$ ，其中 $s_i$ 表示第 $i$ 个域槽名，形如domin-slot，本文使用这种领域名和槽名的组合形式来表示槽名。在第 $t$ 轮的对话状态表示为 $B_t = \{(s_i, v_i^t) | 1 \leq i \leq n\}$ ， $v_i^t$ 表示第 $t$ 轮次第 $i$ 个槽的槽值。对话状态跟踪任务目标就是从对话序列 $P$ 提取出每轮次的对话状态 $B_t$ 。

### 3.2 基于缓解错误传递策略的DLSD-DST模型

将该模型命名为DLSD-DST的原因在于模型使用了对话级状态(Dialogue-level State)和历史对话状态删除策略(Dialogue State Deletion)。该模型的主要思想是两方面：

- 使用对话级状态作为预测目标来增加轮次间预测的独立性，使用槽位状态操作预测和槽值生成框架来维持高效的对话状态跟踪算法计算。对话级状态模型预测目标和轮级状态模型预测目标的示例见表1。
- 在训练中通过一定概率删除上一轮对话状态，使模型不再对上一轮的对话状态的结果产生依赖，使模型对历史对话状态产生怀疑，反而去从历史对话中找到或更正答案，以此来缓解使用历史对话状态产生的错误传递现象。简单示例见表1。

模型的基本结构如图2所示，其中两个子模块的细节则分别在3.3节和3.4节中介绍。

### 3.3 槽位状态操作预测模块

该模块的主要任务是对输入进行编码，预测槽位状态操作符类型。DLSD-DST使用对话级状态进行建模，并以一定的概率删除上一轮对话状态，为了保持信息的完整性，输入的历史对

	第1轮	第2轮	第3轮	第4轮
轮级状态模型预测目标	A,B	C,D	E	F
对话级状态模型预测目标	A,B	A,B,C,D	A,B,C,D,E	A,B,C,D,E,F
轮级状态模型的历史对话状态输入	空	A,B	A,B,C,D	A,B,C,D,E
DLDS-DST模型的历史对话状态输入	空	(A),(B)	A,B,(C),(D)	A,B,C,D,(E)

Table 1: 模型预测目标和历史对话输入的一个简单示例(表中示例使用字母A~F代表每个不同的槽值对, 使用括号圈起来并加粗的字母表示以一定概率删除该值使其不一定出现在输入序列中)

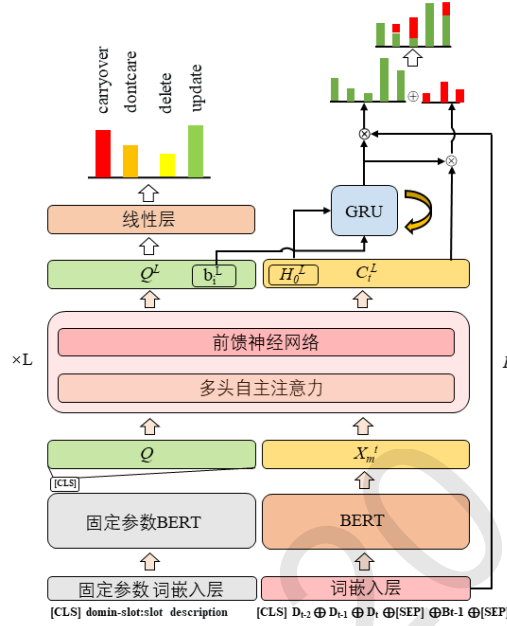


Figure 2: 模型的总体结构

话长度最少为2轮对话。模型在第 $t$ 轮的输入 $M_t$ 如公式(1)所示:

$$M_t = ([cls], D_{t-2}, D_{t-1}, D_t, [sep], B_{t-1}, [sep]) \quad (1)$$

在公式(1)中,  $[cls]$ 是BERT输入序列第一个标识符, 可以将该符号对应的输出向量作为整个输入的语义表示。 $D_t = (u_t, s_t)$ 表示第 $t$ 轮也就是当前轮次的对话。除了当前轮次对话, DLSD-DST还使用之前两轮对话 $D_{t-1}$ 、 $D_{t-2}$ 作为对话历史, 同时使用上一轮的对话状态 $B_{t-1}$ 作为输入来辅助模型进行推理, 在训练时, 模型以一定概率 $\alpha$ 删除上一轮次标签中的轮级对话状态。 $[sep]$ 为分隔符标记。

该模块首先使用BERT(Devlin et al., 2018)作为编码器,对本轮用户对话和历史对话以及上一轮对话状态进行编码。设 $M_t$ 长度为 $m$ ,将 $M_t$ 送入BERT进行编码, 如公式(2)所示:

$$X_m^t = BERT(M_t) = [h_0, \dots, h_i, \dots, h_m] \quad (2)$$

在公式(2)中,  $X_m^t \in \mathcal{R}^{m \times d}$ 是经过BERT编码过后的向量序列,  $h_0 \in \mathcal{R}^d$ 是标记 $[cls]$ 的输出向量,  $d$ 是BERT编码的隐藏层大小,  $h_i \in \mathcal{R}^d$ 是编码输入序列第 $i$ 个标记得到的输出向量。

为了编码槽位表征, 有一个可学习的嵌入矩阵 $Q \in \mathcal{R}^{n \times d}$ 。为了对其初始化, 使用槽位描述和域-槽名组成一个输入序列, 将该序列送入一个不可学习, 参数固定的BERT进行编码, 使用每个输入序列 $[cls]$ 的编码向量作为嵌入矩阵 $Q$ 的初始值。嵌入矩阵 $Q$ 第 $i$ 行向量 $b_i \in \mathcal{R}^d$ 是第 $i$ 个槽位的初始表征向量。

为了更好的表征槽位之间的关联性，DLSD-DST使用Ye et al. (2021)提出的堆叠的槽自注意力机制，与其做法不同，模型将输入编码后的序列向量 $X_m^t$ 与槽位表征矩阵 $Q$ 进行拼接，将拼接后的矩阵向量 $H_t \in \mathcal{R}^{(n+m) \times d}$ 一起输入堆叠的多头自注意力层。

具体来说，堆叠的多头自注意力层由 $L$ 个相同的层组成，每一层有两个子层组成。第一个子层使用自注意力机制(Vaswani et al., 2017)学习槽位表征和输入间的关系，第二个子层是前馈网络(FFN)，每个子层的主要功能前面都有层归一化(Ba et al., 2016)，后面还接有一个残差连接(He et al., 2016)。如以下公式(3)和公式(4)所示：

$$\tilde{h}_t^{l-1} = LayerNorm(H_t^{l-1}) \quad (3)$$

$$H_t^l = FFN(MultiHead(\tilde{h}_t^{l-1}, \tilde{h}_t^{l-1}, \tilde{h}_t^{l-1})) + \tilde{h}_t^{l-1} \quad (4)$$

在公式(4)中， $H_t^l$ 表示第 $l$ ( $1 \leq l \leq L$ )层的输出，最终在第 $L$ 层得到堆叠计算后的输出 $H_t^L \in \mathcal{R}^{(n+m) \times d}$ 。矩阵主要有两部分组成，一部分是槽位表征矩阵 $Q$ 的序列 $[b_1, \dots, b_i, \dots, b_n]$ 的编码 $[b_1^L, \dots, b_i^L, \dots, b_n^L]$ ，代表各个槽门最终的向量表征；另一部分是 $X_m^t$ 的编码 $C_t^L = [h_0^L, \dots, h_i^L, \dots, h_m^L] \in \mathcal{R}^{m \times d}$ ，代表输入序列的表征向量。

对于每个槽位的状态操作预测执行四分类，该做法与Kim et al. (2020)一致，如公式(5)所示：

$$P_{op,t}^i = Softmax(W_{op} b_i^L) \quad (5)$$

在公式(5)中， $W_{op} \in \mathcal{R}^{4 \times d}$ 是可学习的参数矩阵，将槽位表征 $b_i^L$ 进行线性变换， $P_{op,t}^i \in \mathcal{R}^4$ 是经过softmax函数得到的槽位操作状态的概率分布。每个槽位有四种状态预分别是{None,DELETE,DONTCARE,UPDATE}，None表示不更生成该槽位的值，DELETE表示将该槽位原有的预测删除，DONTCARE表示该槽位值设置为不关心特殊字符，UPDATE表示使用槽值生成器生成槽值。模型只将分类为UPDATE的槽位表征向量送入槽值生成模块进行解码。

### 3.4 槽值生成模块

DLSD-DST使用与Kim et al. (2020)一致的槽位生成机制，该机制采用soft-copy(See et al., 2017)进行生成。对于每一个分类是UPDATE的槽位表征 $b_i^L$ ，使用GRU循环神经网络(Cho et al., 2014)进行解码。使用[cls]的表征向量 $h_0^L$ 对GRU的隐藏层状态 $g_i^0$ 进行初始化，使用 $b_i^L$ 作为GRU解码器的初始化输入 $r_i^0$ ，循环更新隐藏层状态直到得到[EOS]终止表示符来终止计算。循环神经网络的计算如公式(6)所示：

$$g_i^k = GRU(g_i^{k-1}, r_i^{k-1}) \quad (6)$$

在公式(6)中， $k$ 表示循环解码的步数。模型在第 $k$ 步进行解码时，使用当前步数计算的隐藏层 $g_i^k$ 与BERT预训练模型的文本嵌入层矩阵 $E \in \mathcal{R}^{V \times d}$ ，先计算一个词汇表的概率分布，其中 $V$ 表示BERT嵌入层的词表大小。如公式(7)所示：

$$P_{i,vcb}^k = softmax(Eg_i^k) \in \mathcal{R}^V \quad (7)$$

模型再与输入序列的表征向量 $C_t^L$ 和隐藏层 $g_i^k$ 计算一个上下文的概率分布，最后将上下文的概率分布和词汇表的概率分布对应相加，如公式(8)和公式(9)所示。

$$P_{i,ctx}^k = softmax(C_t^L g_i^k) \in \mathcal{R}^m \quad (8)$$

$$P_i^k = \theta P_{i,ctx}^k + (1 - \theta) P_{i,vcb}^k \in \mathcal{R}^V \quad (9)$$

公式(9)中的 $\theta$ 是个标量，用来平衡词汇表概率分布和上下文概率分布，由以下公式计算：

$$\theta = sigmod(W_1 [g_i^k; r_i^k; c_i^k]) \quad (10)$$

在公式(10)中， $W_1 \in \mathcal{R}^{1 \times 3d}$ 是个可学习参数， $c_i^k = P_{i,ctx}^k C_t^L \in \mathcal{R}^d$ 表示上下文向量。最终输出单词选取概率分布 $P_i^k$ 中最大的一个。

### 3.5 损失函数

槽位状态操作分类使用交叉熵损失函数，所有 $n$ 个槽位的损失如公式(11)所示：

$$L_{op} = -\frac{1}{i} \sum_{i=1}^n (Y_{op,t}^i)^\top \log (P_{op,t}^i) \quad (11)$$

对于槽值生成，DLSD-DST同样使用交叉熵损失函数，注意模型只需要更新 $n_1$  ( $0 \leq n_1 \leq n$ )个槽,如公式(12)所示：

$$L_{gen} = -\frac{1}{n_1} \sum_{i \in n_1} \left[ \frac{1}{K_{gen}} \sum_{k=1}^{K_{gen}} (Y_i^k)^\top \log (P_i^k) \right] \quad (12)$$

其中 $K_{gen}$ 是该槽值应该更新的值的长度，对应GRU应该循环的次数。DLSD-DST联合训练槽位状态操作分类模块和槽值生成模块,总损失如公式(13)所示：

$$loss = L_{gen} + L_{op} \quad (13)$$

## 4 实验设置

### 4.1 数据集设置

本文使用MultiWOZ 2.1(Eric et al., 2020)和MultiWOZ 2.4(Ye et al., 2022)作为实验中的数据集。MultiWOZ 2.0(Budzianowski et al., 2018)是最具挑战性的多域任务对话数据集之一，是任务型对话研究中广泛使用的数据集。数据集总共有超过10000个任务型对话，每个对话轮次长短不一，既有很长的涉及多领域的对话，也有较短的单领域对话。MultiWOZ 2.1是MultiWOZ 2.0的改进版本，其更正了部分注释错误，并加入了对域槽的描述。MultiWOZ 2.4是因2.1版本的测试集标注中噪声非常多导致各种DST模型评估受到太多错误的噪声影响而改进的。其在MultiWOZ 2.1基础上的进行改进,其训练集则保持不变，验证集和测试集中的注释错误已被手动更正，以用来更好的进行评估。为了方便与其他模型进行评估，模型使用与TRADE(Wu et al., 2019)一致的预处理过程,只使用了5个领域(餐厅、火车、酒店、出租车、景点)，不包括医院域和警察域。

### 4.2 评价指标

本文计算所有测试集上的联合目标精度(joint goal accuracy,JGA),联合目标精度是所有槽的值都被正确预测的对话轮次占总轮次的比例。JGA是DST任务中最重要的度量。槽精度(slot accuracy)定义为所有单独槽位精度的平均值。每个槽位的准确性是根据其值被正确预测的对话轮次的比率来计算的。

### 4.3 训练设置

DLSD-DST使用bert-base-uncased(Devlin et al., 2018)作为用来编码的预训练模型，隐藏向量的大小为768维，最大序列长度为512。self-attention层的层数设置为10，注意力头数为4，槽位操作预测分类数为4，类别是{None,DELETE,DONTCARE,UPDATE}。生成器隐藏向量大小一样是768维，生成器的令牌嵌入矩阵与槽位操作预测器的令牌嵌入矩阵共享参数。由于使用了预训练模型，DLSD-DST将槽位操作预测器和生成器分别设置学习率以加速收敛。槽位操作预测器的峰值学习率设置为 $4e-5$ ，生成器的峰值学习率设置为 $1e-4$ ,两个模块都使用0.1的学习率热身比例。训练使用AdamW(Loshchilov and Hutter, 2017)优化器优化，训练的batch size设置为30，dropout概率设置为0.1。DLSD-DST使用了字词掩码，以0.1的几率将输入的令牌替换成[UNK]令牌。训练时DLSD-DST以0.8的概率随机删除对话状态。为了防止模型对对话状态的顺序产生依赖，DLSD-DST随机将对话状态序列的顺序打乱。DLSD-DST联合训练槽位操作预测器和槽值生成器30个epoch，并报道在测试集表现最佳的模型。

## 5 实验结果与分析

### 5.1 模型性能对比

本文将与近期基于开放词汇表的先进DST方法进行比较。为了更好的对比结果，本文使用BASELINE模型作为对比，该模型使用轮级状态作为预测目标，使用完整的历史对话状态作为输入，其余参数和DLSD-DST设置一样。

所得实验结果如表2所示。可以看出，DLSD-DST在MultiWOZ 2.4上获得了具有竞争性的结果，但在MultiWOZ 2.1上表现并不出彩，只比SOM-DST好上1.32%。DLSD-DST获得了和SOM-DST较为一致的结果，可能是因为DLSD-DST使用同样的模型架构，并在数据集标签修正中获得了更多增益，这种增益是测试集的标签修正结果，并不是训练集的修正增益，MultiWOZ 2.1的测试集中的一些错误标签，影响了模型的评估。

Models	MultiWOZ 2.1	MultiWOZ 2.4
SimpleTOD (Budzianowski and Vulić, 2019)	51.75%	57.18%
TripPy (Heck et al., 2020)	55.18%	60.55%
SOM-DST (Kim et al., 2020)	51.24%	66.78%
SAVN (Wang et al., 2020)	54.86%	60.55%
Seq2seq (Zhao et al., 2021)	54.40%	67.10%
TripPy-R (Heck et al., 2022)	<b>55.99%</b>	69.87%
BASELINE	51.69%	67.60%
DLSD-DST(ours)	52.56%	<b>70.95%</b>

Table 2: 模型在MultiWOZ 2.1和MultiWOZ 2.4测试集上的联合目标精度

### 5.2 特定领域精度

特定领域精度是在预测对话状态的子集上计算的，子集由特定于某个域的所有槽位组成，计算这些域槽位的联合目标精度即可得到特定领域精度。本节测试了模型在MultiWOZ 2.4数据集上特定领域的精度如表3所示。可以看见，本文提出的DLSD-DST模型相较于BASELINE模型在各个领域的准确度都获得了提升，并在hotel领域，获得了较大的性能提升。可见，DLSD-DST模型对于baseline模型是全方位的提高，并不是仅仅通过增强一两个领域而获得的。

Domain	som-dst	BASELINE	DLSD-DST
hotel	62.07%	64.91%	<b>71.70%</b>
train	82.76%	82.86%	<b>83.50%</b>
restaurant	77.56%	78.90%	<b>80.97%</b>
attraction	79.96%	82.27%	<b>83.46%</b>
taxi	65.21%	63.34%	<b>65.37%</b>

Table 3: 模型在MultiWOZ 2.4各个特定领域的精度及模型的槽精度

序号	真实操作符标签	真实上一轮对话状态标签	真实槽值标签	SOM-DST	+DLSD	BASELINE	+DL	+DLSD
1	×	×	×	66.49%	<b>69.16%</b>	67.60%	68.61%	<b>70.95%</b>
2	×	✓	×	<b>89.07%</b>	87.08%	<b>89.40%</b>	88.97%	87.16%
3	×	✓	✓	<b>90.36%</b>	89.96%	90.73%	<b>92.68%</b>	90.80%
4	✓	✓	×	<b>97.80%</b>	96.35%	<b>97.95%</b>	96.73%	97.03%
5	✓	×	✓	100%	100%	100%	100%	100%
6	×	×	✓	70.02%	<b>74.32%</b>	71.35%	76.43%	<b>77.63%</b>
7	✓	×	×	<b>93.66%</b>	90.97%	<b>94.26%</b>	88.56%	89.43%

Table 4: DLSD方法在MultiWOZ 2.4测试集上的联合目标精度。DL表示使用对话级状态为预测目标，SD表示使用随机删除上一轮对话状态机制

### 5.3 消融实验

DLSD-DST获得了良好的性能表现，为了证明这种性能提升来自哪里，设计如下消融实验，如表4所示。一共7组消融实验，分别对BASELINE和DLSD-DST在不同设置下进行性能测试，测试的指标是联合目标精度。实验分别使用了以下三种设置：

- 使用真实的操作符标签替代模型自己预测的操作符标签，这将使模型在测试中获得正确预知槽位操作的能力
- 使用真实的之前轮次的对话状态标签替代上一轮模型自己生成的，这将消除模型轮次间的错误传递
- 使用真实的槽值标签替代模型生成的槽值，这样可以排除生成器的干扰

在序号2的实验中，BASELINE在使用真实对话状态标签的情况下精度比DLSD-DST的精度更高，结果说明了在消除轮次传递误差的情况下，DLSD-DST并不比BASELINE好。进一步验证结果，在序号3和序号6的实验中，在槽位操作符预测任务上，DLSD-DST仅有微弱优势，虽然序号6有较大的性能进步，但是这可能来自于轮次间的错误修正。在序号4和序号7的实验中，DLSD-DST的槽值生成器性能要比BASELINE模型弱很多，这可能是过拟合造成的。通过以上实验可以看出，是否使用上一轮次的真实对话状态标签成为了影响性能的主要因素，DLSD-DST显然从轮次之间的错误修正中获得了更多增益。

为了进一步验证所得的结论，在SOM-DST上应用了DLSD方法，如表4所示，唯一的不同是在序号3的实验，SOM-DST的槽位操作符预测精度要高一些。以上实验都说明模型在应用了本章提出的DLSD方法后会从轮次间的错误修正获得更多增益。

### 5.4 各组件对模型的影响

进一步探讨DLSD方法的各个组件对模型的影响。如表4所示，两个DL和SD组件都使模型的性能得到了提高。对使用了DL组件的模型进行评估，发现其槽位预测任务和槽值生成任务在也消除误差传递的情况下表现不如BASELINE，但却取得了性能上的超越，这点和DLSD-DST一致，这说明使用DL组件后获得了一定错误修正增益。进一步分析发现，其槽位操作符预测任务表现要好于DLSD-DST和BASELINE，这部分性能提高可能来源于样本数量不平衡的缓解。槽位操作符分类是一个各样本不平衡的分类任务，大多数槽位都是不更新的，往往只有少量样本需要更新，使用DL组件的时候，需要更新的样本数量变多了，缓解了样本的不平衡，提高了小样本分类的准确性，所以观察到了槽位操作分类任务性能的提高。在进行使用对话级状态为预测目标的时候，获得了来自槽位分类精度的提升和部分误差传递缓解的增益，但是槽值生成器可能是因为过拟合而出现了性能下降。

如表4所示，在DL的基础上增加了HR组件后，模型的槽位操作符分类器的精度出现下降，槽值生成器的性能出现了一定程度的回升，在去除轮次间传递误差因素以后，模型性能小幅度下降，这说明了模型从轮次间的错误传递的缓解中获得了更多增益。当以一定概率删除之前轮次的对话状态后，模型会进一步减轻对历史对话状态的信任，从而去查阅历史对话数据，将历史对话数据和历史对话状态进行综合分析，从而起到了及时更正错误的作用，缓解了错误的传递的现象。

### 5.5 模型的修正行为

为了进一步验证结果，对DLSD-DST的结果进行统计分析。如果模型在当前轮次预测错误，并能在后续轮次中正确预测，那么就认为此次正确预测是一次修正行为。实验测试了BASELINE模型和DLSD-DST模型在测试集上的修正行为次数，如表5所示。可以看出，BASELINE模型在应用了DLSD方法后，修正次数增长了45.03%，而性能增长了4.96%，这说明DLSD方法可以使模型产生更多修正行为，缓解了轮次间的错误传递。

	BASELINE	DLSD-DST
修正行为次数	131	190

Table 5: 模型在MultiWOZ 2.4测试集上的修正行为次数

### 5.6 超参数的设置

DLSD-DST使用了堆叠的自注意力层，这个层数的数目影响着模型的性能。选取{4,6,8,10,12}作为实验的层数选取范围，在丢弃率 $\alpha$ 为0.5情况下，观察到模型性能从4到10性能在逐层递增，并在第10层获得最佳性能，如图3所示。当层数L超过一定数量时，模型性能出现一定下降，可能是由于过拟合所致。

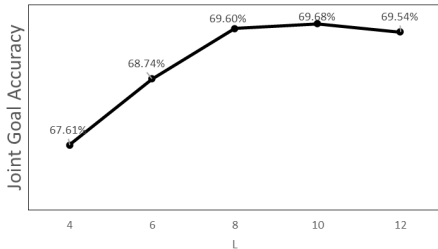


Figure 3: 自注意力层的层数对模型的影响

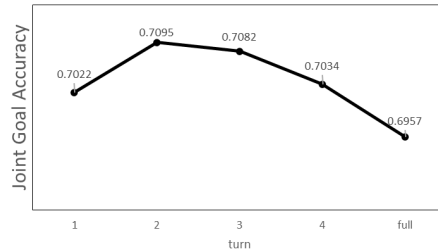


Figure 4: 历史对话轮次的长度对模型的影响

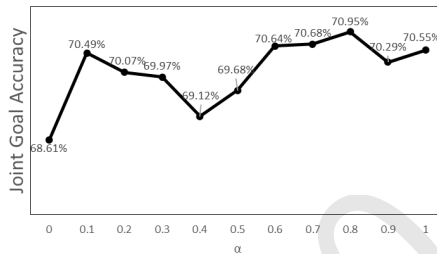


Figure 5: 历史状态删除概率 $\alpha$ 对模型的影响

由于模型以一定概率 $\alpha$ 对上一轮次的对话状态进行删除，过短的历史对话可能造成信息的缺失，所以探究历史对话轮次对模型性能的影响很有必要。分别选取{1,2,3,4,full}作为实验的超参数进行测试,所得结果如图4所示。可以看到，在仅仅使用2轮对话历史的时候，模型性能到达最佳。在2轮次之后，过长的历史对话导致了模型性能的下降。模型使用删除上一轮次对话状态的机制，引导模型回顾之前轮次的对话历史，可能使模型对前几轮历史数据更加敏感，而之后过长的历史对话导致了数据的稀疏性，影响了模型性能。

对于上一轮对话状态删除概率 $\alpha$ 的设置，选取测试了从0到1，以0.1为间隔的值，如图5所示。可以看出，并未发现超参 $\alpha$ 对模型有有规律的影响，如果要获得最佳的性能表现，需要进行实验测试而选定。当删除概率为1时，模型永远无法知道上一轮次的预测结果，当删除概率为0时，模型退化为普通对话级状态模型。实验出现较大波动的原因和模型输入的历史对话数量，数据集的分布情况有关，还需要在后续中的工作中进行分析。

## 6 Conclusions

本文提出了DLSD-DST模型，该模型以对话级状态为预测目标，并一定概率删除上一轮对话状态，迫使模型在存在不完全可信的对话状态情况下，主动回顾历史对话信息，纠正错误的对话状态，缓解了错误传递对模型造成的影响。基于DLSD方法提出的DLSD-DST模型在MultiWOZ 2.4取得了具有竞争力的结果。DLSD-DST同样延续了SOM-DST模型高效的架构，一次抽取所有槽值，不需要重头对每个槽进行单独生成，不依赖设置固定的槽值列表，而是利用复制机制生成槽值。而相对于以往的全历史模型，DLSD-DST仅仅使用2轮次历史对话和部分对话状态，输入的规模更低，减少了计算开销。实验显示模型超参数中的历史对话状态删除概率 $\alpha$ 存在选取困难的问题，难以测试找到最佳的值，希望能在以后的工作中引用元学习方法对该参数的选取进行优化。

## 参考文献

- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *stat*, 1050:21.
- Vevake Balaraman, Seyedmostafa Sheikhalishahi, and Bernardo Magnini. 2021. Recent neural methods on dialogue state tracking for task-oriented dialogue systems: A survey. In *Proceedings of the 22nd annual meeting of the special interest group on discourse and dialogue*, pages 239–251.
- Paweł Budzianowski and Ivan Vulić. 2019. Hello, it’s gpt-2-how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.
- H. Chen, X. Liu, D. Yin, and J. Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *8th Workshop on Syntax, Semantics and Structure in Statistical Translation, SSST 2014*, pages 103–111. Association for Computational Linguistics (ACL).
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 422–428.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44.
- Michael Heck, Nurul Lubis, Carel van Niekerk, Shutong Feng, Christian Geishauser, Hsien-Chin Lin, and Milica Gasic. 2022. Robust dialogue state tracking with weak supervision and sparse data. *Transactions of the Association for Computational Linguistics*, 10:1175–1192.
- A Yeong Kim, Hyun Je Song, and Seong Bae Park. 2018. A two-step neural dialog state tracker for task-oriented dialog processing. *Computational Intelligence and Neuroscience*, 2018:1–11.
- Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582.
- Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy, July. Association for Computational Linguistics.
- Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. Mintl: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

- Ryuichi Takanobu, Qi Zhu, Jinchao Li, Baolin Peng, Jianfeng Gao, and Minlie Huang. 2020. Is your goal-oriented dialog model performing really well? empirical analysis of system-wise evaluation. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 297–310.
- Xin Tian, Liankai Huang, Yingzhan Lin, Siqi Bao, Huang He, Yunyi Yang, Hua Wu, Fan Wang, and Shuqi Sun. 2021. Amendable generation for dialogue state tracking. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 80–92.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Yexiang Wang, Yi Guo, and Siqi Zhu. 2020. Slot attention with value normalization for multi-domain dialogue state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3019–3028.
- Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.
- Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14230–14238.
- Fanghua Ye, Jarana Manotumruksa, Qiang Zhang, Shenghui Li, and Emine Yilmaz. 2021. Slot self-attentive dialogue state tracking. In *Proceedings of the Web Conference 2021*, pages 1598–1608.
- Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2022. Multiwoz 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 351–360.
- Yan Zeng and Jian-Yun Nie. 2020. Jointly optimizing state operation prediction and value generation for dialogue state tracking. *arXiv e-prints*, pages arXiv–2010.
- Jeffrey Zhao, Mahdis Mahdih, Ye Zhang, Yuan Cao, and Yonghui Wu. 2021. Effective sequence-to-sequence dialogue state tracking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7486–7493.
- Su Zhu, Jieyu Li, Lu Chen, and Kai Yu. 2020. Efficient context and schema fusion networks for multi-domain dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 766–781.