基于层次化注意力网络与自适应训练目标的对话状态跟踪

本文基于 ACL-2020 论文《A Contextual Hierarchical Attention Network with Adaptive Objective for Dialogue State Tracking》，论文由中国科学院计算技术研究所和微信 AI 团队合作完成。

导语：在对话状态跟踪（Dialogue State Tracking）任务中，模型通常需要利用对话历史来确定每轮对话的对话状态（即多个槽值对的集合）。但是，现有方法在建模槽与对话历史之间的交互时存在一些不足，并限制了模型从对话历史中捕获相关信息的能力。针对此问题，作者提出一种 Contextual Hierarchical Attention Network（CHAN）模型，分别在词、对话轮次这两个级别建模槽与对话历史之间的交互，并使用一个独立的上下文编码器对各轮次相关信息之间的关系进行编码。此外，在对话状态跟踪任务中，每个槽的出现频次存在不均衡的现象，这使得每个槽的学习难度各不相同并限制了模型的整体性能，现有方法尚未对此进行探索。针对这个槽不均衡问题，作者提出一种自适应训练目标（Adaptive Objective），在训练过程中动态地评估每个槽的学习难度并自适应地调整其学习权重。作者在两个大型对话数据集上开展了实验，提升显著，并取得当前最佳效果（State-of-The-Art）。

一、 模型背景与简介

近年来，任务型对话系统（例如腾讯小微）取得了长足的进展。对话状态跟踪是任务型多轮对话系统中的一个重要模块，给定当前轮次的对话和对话历史，对话状态跟踪在每个轮次输出相应的对话状态（即多个槽值对的集合）。

在对话状态跟踪任务中，对话状态通常依赖于对话历史中的相关信息，如图 1 所示，第 3 轮的对话状态“price range=moderate, food=Chinese”依赖于第 2 轮对话。然而，传统的对话状态跟踪模型通常只考虑当前轮次的对话、并没有利用对话历史(如 GLAD 模型)，后来的研究工作尝试在对话状态跟踪中引入对话历史，但是这些方法在建模槽与对话历史之间的交互时存在一些不足，比如：仅对拼接后的历史对话语句使用简单的注意力机制来获取相关信息（TRADE 等模型）；未利用到完整的对话历史（COMER 等模型）；槽与对话历史缺少直接的交互（StateNet 等模型）；等等。这些不足限制了这些方法从对话历史中捕获相关信息的能力。

User: Hello, I'm looking for a resraurant with fair prices. **State:** *price range=moderate*
Sys: OK. There are Golden Wok Chinese restaurant and Nirala which serves Indian food, which one do you like? **User:** Are they both have a reasonable price ? **State:** *price range=moderate*
Sys: Of course. **User:** Please tell me the address of Golden Wok. **State:** *price range=moderate; food=chinese*

图 1：对话示例。在第 3 轮对话中，对话状态（用斜体表示）依赖于对话历史中的相关信息（用下划线表示），“User”和“Sys”分别表示人类和机器。

此外，不同槽或槽值对的出现频次具有差异性。例如，在 MultiWOZ 2.0 数据集中，槽“train-day”共出现 15384 次，而“attraction-name”只有 5843 次；槽值对（attraction-area, center）共出现 5432 次，而（taxi-departure, royal spice）只有 9 次。作者将这个问题描述为槽不均衡问题，如图 2 所示，频次的不同导致了不同槽或槽值对的准确率存在差异，总体上，出现频次越高，准确率越高，这导致有些槽难以学习并限制了对话状态跟踪的整体性能。

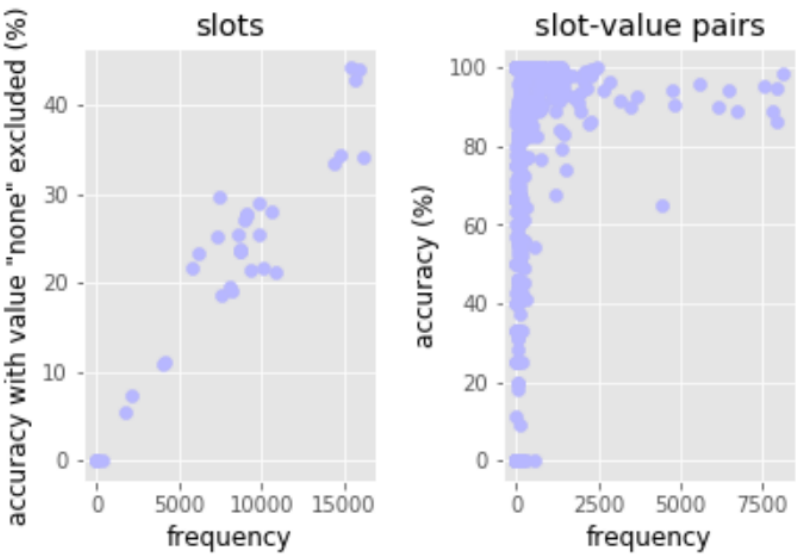

图 2：槽不均衡问题。总体上，出现频次越高，准确率越高。

为了解决以上两个问题，作者分别提出一种 Contextual Hierarchical Attention Network（CHAN）和一种自适应训练目标（Adaptive Objective）用于高效地从对话历史中捕获相关信息，以及缓解槽不均衡问题。在 CHAN 中，槽首先从每个轮次的对话中检索出词级别的相关信息，然后上下文编码器对各轮次相关信息进行语境化编码，最后槽再将这些语境化编码聚合成对话轮次级别的相关信息。为进一步提升 CHAN 从对话历史中捕获相关信息的能力，作者使用一个 State Transition Prediction 任务用来与对话状态跟踪任务联合训练。针对槽不均衡问题，自适应训练目标可以动态地评估每个槽的学习难度并自适应地调整其学习权重，从而尽可能平衡不同槽之间的学习。作者在两个大型对话数据集：MultiWOZ 2.0 和 MultiWOZ 2.1 上开展了实验，均获得显著提升，并且取得当前最佳效果（State-of-The-Art）。这个工作的贡献点如下：

1. 作者提出了一个 Contextual Hierarchical Attention Network 用于从对话历史中高效地捕获相关信息，并使用 State Transition Prediction 任务进一步增强捕获相关信息的能力。

2. 作者设计了一个自适应训练目标用于解决槽不均衡问题。这个工作是对话状态跟踪任务中首个关注槽不均衡问题的工作。

3. 实验结果表明，模型在多个数据集上提升显著并获得 State-of-The-Art 效果。

二、 模型结构

模型将对话状态跟踪任务建模成了一个多标签分类问题，给定一个 T 轮对话 $X=\{(U_1,R_1),\ldots,(U_T,R_T)\}$，其中 U_t 和 R_t 分别表示第 t 轮的用户语句和系统回复，那么第 t 轮的对话状态为 $B_t=\{s, v_t, s\in S\}$，其中 S 是所有槽的集合，v_t 表示槽 s 对应的值。图 3 给出了模型的整体架构。

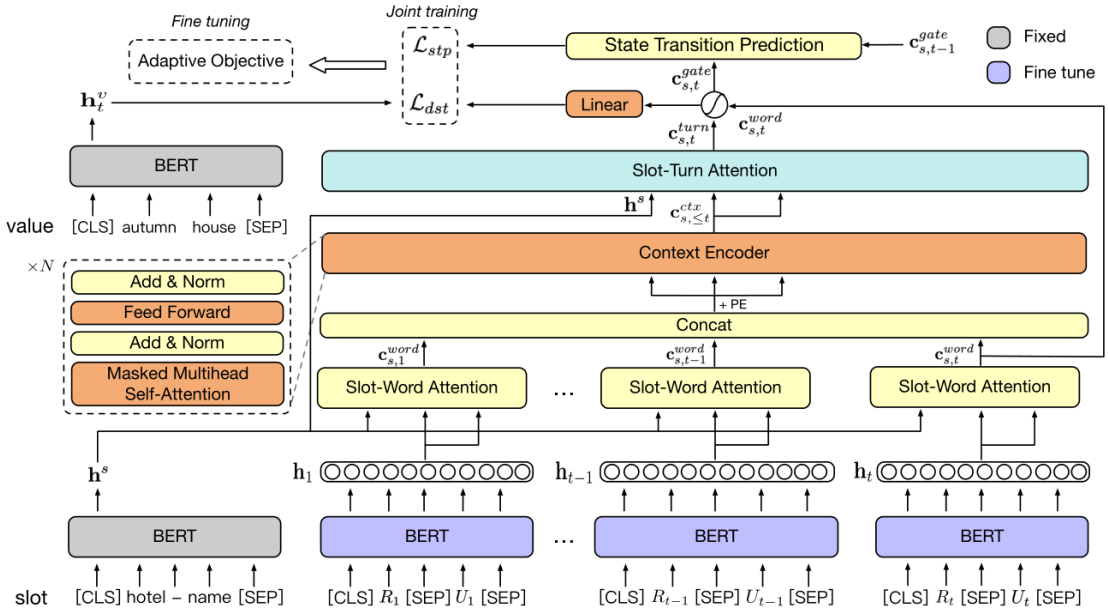

图 3：模型架构

如图 3 所示，模型使用预训练语言模型 BERT 作为底层编码单元。在 BERT 中，[CLS]和[SEP]分别用于表示句子的整体含义和句子分隔符。对于同一个轮次内的系统回复 R_t 和用

户语句 U_t，作者将它们拼接起来视为一个句子，并使用 $BERT_{finetune}$（在训练时 BERT 参数可调）对其进行编码，得到每个词的深层表示：

$$\mathbf{h}_t = \mathrm{BERT}_{finetune}([R_t; U_t])$$

对于槽 s 和值 v_t，作者将他们分别视为一个句子，使用 $BERT_{fix}$（在训练时 BERT 参数被固定住）对其编码，并使用[CLS]位置的表示作为整个句子的表示：

$$\begin{aligned}\mathbf{h}^s &= \mathrm{BERT}_{fixed}(s)\\ \mathbf{h}_t^v &= \mathrm{BERT}_{fixed}(v_t)\end{aligned}$$

然后，模型开始进行槽与对话历史之间的交互。在词级别，模型使用一个 Multi-Head Attention 用于在每个轮次中捕获词级别的相关信息：

$$\mathbf{c}_{s,t}^{word} = \mathrm{MultiHead}(\mathbf{h}^s, \mathbf{h}_t, \mathbf{h}_t)$$

模型使用一个 N 层的单向 Transformer 单元作为上下文编码器，用于编码{1,...t}轮词级别相关信息之间的交互，其中每一层由一个 Masked Multi-Head Self Attention 和一个 Position-Wise Feed-Forward Network 组成。m^n 表示第 n 层的输出，PE 表示 Positional Encoding，整个过程可定义为：

$$\begin{aligned}\mathbf{m}^n &= \mathrm{FFN}(\mathrm{MultiHead}(\mathbf{m}^{n-1}, \mathbf{m}^{n-1}, \mathbf{m}^{n-1}))\\ \mathbf{m}^0 &= [\mathbf{c}_{s,1}^{word} + \mathrm{PE}(1), ..., \mathbf{c}_{s,t}^{word} + \mathrm{PE}(t)]\\ \mathbf{c}_{s,\leq t}^{ctx} &= \mathbf{m}^N\end{aligned}$$

在对话轮次级别，模型也使用一个 Multi-Head Attention 用于在整个对话中捕获轮次级别的相关信息：

$$\mathbf{c}_{s,t}^{turn} = \mathrm{MultiHead}(\mathbf{h}^s, \mathbf{c}_{s,\leq t}^{ctx}, \mathbf{c}_{s,\leq t}^{ctx})$$

特别地，模型使用一个 Global-Local Fusion Gate 机制用于平衡整个对话的相关信息（Global）和第 t 轮的局部对话信息（Local）：

$$g_{s,t} = \sigma(\mathbf{W}_g \odot [\mathbf{c}_{s,t}^{word}; \mathbf{c}_{s,t}^{turn}])$$

$$\mathbf{c}_{s,t}^{gate} = g_{s,t} \otimes \mathbf{c}_{s,t}^{word} + (1 - g_{s,t}) \otimes \mathbf{c}_{s,t}^{turn}$$

最后，模型使用一个线性变换得到最终的输出特征，计算特征与各候选值 v_t 之间的 L2 距离，并使用交叉熵作为损失函数：

$$\mathbf{o}_{s,t} = \mathrm{LayerNorm}(\mathrm{Linear}(\mathrm{Dropout}(\mathbf{c}_{s,t}^{gate})))$$

$$p(v_t|U_{\leq t}, R_{\leq t}, s) = \frac{\exp(-\|o_{s,t} - h_t^v\|_2)}{\sum_{v' \in \mathcal{V}_s} \exp(-\left\|o_{s,t} - h_t^{v'}\right\|_2)}$$

$$\mathcal{L}_{dst} = \sum_{s \in \mathcal{S}} \sum_{t=1}^{T} -\log(p(\hat{v}_t|U_{\leq t}, R_{\leq t}, s))$$

作者使用 State Transition Prediction 任务用来进一步增强模型捕获相关信息的能力，这个任务是用来预测第 t 轮的槽值对相比第 t-1 轮而言是否发生改变。作者先将输出特征映射到对应的特征空间，然后将相邻轮次的特征作为输入得到第 t 轮槽值对发生改变的概率，最后使用交叉熵计算损失：

$$\mathbf{c}_{s,t}^{stp} = \tanh(\mathbf{W}_c \odot \mathbf{c}_{s,t}^{gate})$$

$$p_{s,t}^{stp} = \sigma(\mathbf{W}_p \odot [\mathbf{c}_{s,t}^{stp}; \mathbf{c}_{s,t-1}^{stp}])$$

$$\mathcal{L}_{stp} = \sum_{s \in \mathcal{S}} \sum_{t=1}^{T} -y_{s,t}^{stp} \cdot \log(p_{s,t}^{stp})$$

对于自适应训练目标，作者受到了目标检测中 Focal Loss 的启发，使用 Soft Sampling 的方式对不同槽的学习权重进行调整，不同点在于：Focal Loss 为不同的槽设置一个静态的学习权重、在整个训练过程中不再发生变化，而且需要人工调参；而自适应训练目标在训练过程中会自适应地计算不同槽的学习权重并动态地对其进行调整、不引入新的超参数，可以更好地拟合数据并取得更优的性能。具体地，对于每个槽，自适应训练目标将槽在验证集上的准确率（$\mathrm{acc}_s^{\mathrm{val}}$）作为学习难度的指标，并由此得出一个归一化的训练权重，使得那些学习难度更大的槽获得一个更大的学习权重，从而鼓励那些困难槽的学习并尽可能平衡不同槽的训练。在每个训练轮次结束时，自适应训练目标都会重新评估学

习难度并更新对应的学习权重，所以学习权重不仅是随训练过程动态改变的，也是根据学习难度自适应计算得到的。对于每个样本{(U_t, R_t), (s, v_t)}，p(s, v_t)是样本的置信度。整个过程如下：

$$\alpha_s = \frac{1 - acc_s^{val}}{\sum_{s' \in \mathcal{S}} 1 - acc_{s'}^{val}} \cdot |\mathcal{S}|$$

$$\beta(s, v_t) = (1 - p(s, v_t))^{\gamma}$$

$$\mathcal{L}_{adapt}(s, v_t) = -\alpha_s \beta(s, v_t) \log(p(s, v_t))$$

模型的训练过程分为两步。第一步，将对话状态跟踪任务和 State Transition Prediction 任务联合训练：

$$\mathcal{L}_{joint} = \mathcal{L}_{dst} + \mathcal{L}_{stp}$$

第二步，使用自适应训练目标，在对话状态跟踪任务上对模型进行精调：

$$\mathcal{L}_{finetune} = \sum_{s \in \mathcal{S}} \sum_{t=1}^{T} \mathcal{L}_{adapt}(s, \hat{v}_t)$$

三、 实验结果

作者在两个大型对话数据集 MultiWOZ 2.0 和 MultiWOZ 2.1 上进行了实验，使用 Joint Accuracy（对话状态的准确度）和 Slot Accuracy（槽值对的准确度）作为评价指标。结果如下：

Model	Ontology	MultiWOZ 2.0		MultiWOZ 2.1	
		Joint (%)	Slot (%)	Joint (%)	Slot (%)
DSTreader (Gao et al., 2019b)	×	39.41	-	36.40⋆	-
GLAD-RCFS (Sharma et al., 2019)	✓	46.31	-	-	-
HyST (Goel et al., 2019)	✓	42.33	-	38.10⋆	-
TRADE (Wu et al., 2019)	×	48.60	96.92	45.60⋆	-
DST-QA (Zhou and Small, 2019)	×	51.44	97.24	51.17	97.21
SOM-DST (Kim et al., 2019)	×	51.38	-	52.57	-
SUMBT (Lee et al., 2019)	✓	48.81†	97.33†	52.75‡	97.56‡
DST-picklist (Zhang et al., 2019)	✓	-	-	53.30	-
Our Model	✓	**52.68**	**97.69**	**58.55**	**98.14**

图 4：在 MultiWOZ 2.0 和 MultiWOZ 2.1 上的实验结果

如图 4 所示，在两个数据集上，模型均取得了最佳效果，提升显著。作者发现模型在 MultiWOZ 2.1 上相比 MultiWOZ 2.0 提升更大，这可能是因为 MultiWOZ 2.1 中修复了许多标注错误，模型获益更大。

作者开展了消融实验用于评估各模块的效果。如图 5 所示，CHAN、State Transition Prediction、自适应训练目标都是有效的。对比 Focal Loss，自适应训练目标取得了+0.45 的提升，证明了它可以通过动态评估不同槽的学习难度并自适应地计算学习权重来更好地拟合数据，从而取得更高的性能。

Model	MultiWOZ 2.1
Our Model	**58.55**
- state transition prediction	57.86 (-0.69)
- adaptive objective fine-tuning	57.45 (-1.10)
- above two (only CHAN)†	57.00 (-1.55)
Our Model (FL (α=1,γ=2))‡	58.10 (-0.45)

图 5：消融实验

除此之外，作者还对词、对话轮次两个级别的 Attention 进行了可视化。如图 6 所示，模型在第 5 轮中预测槽“restaurant-name”对应的值时，必须从对话历史中检索与槽有关的信息才能得出正确的预测（“dojo noodle bar”）。其中，相关信息主要分布在第 3 轮与

第 4 轮（红色高亮标记）。从热力图可以看出：在对话轮次级别，注意力权重主要集中在第 3 轮和第 4 轮；在词级别，注意力权重主要集中在第 3 轮与第 4 轮中“dojo noodle bar”分布的位置。这证明了模型能有效利用对话历史并分别从词、对话轮次两个级别捕获与槽相关的信息。

图 6：Attention 可视化

四、总结

本文提出了一个用于对话状态跟踪的新模型，它包含一个 Contextual Hierarchical Attention Network 和一个自适应训练目标分别用于改善对对话历史中相关信息的建模和缓解槽不均衡问题。模型在两个大型对话数据集上提升显著并取得了最佳效果。虽然模型基于 Predefined Ontology，但模型在面临新的槽或值时依然具有良好的可扩展性和通用性。这个工作中所提出的方法也可以用于基于 Open Vocabulary 的模型，作者将在未来进行更深入的探索。