LS-DST: Long and Sparse Dialogue State Tracking with Smart History Collector in Insurance Marketing

Liqiang Song   Mengqiu Yao   Ye Bi
Zhenyu Wu   Jianming Wang   Jing Xiao
Ping An Technology Shenzhen Co., Ltd
Juan Wen   Xin Yu
Ping An Life Insurance Company

Speaker: **Ye Bi**

About the Speaker

- Senior Algorithm Engineer in Ping An Technology Shenzhen Co., Ltd
- **Major Research Direction:** Deep Learning, Recommendation System, Knowledge Graph

Online Insurance is Tendency

Online insurance is becoming more and more popular, though it is in its growth bottleneck. Automatically tracking customers' states can effectively remove the heavy burdens of insurance agents, making them more deliberate to communicate with their customers.

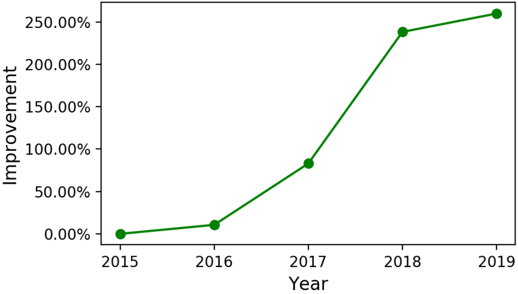

Figure 1: The Number of Insured Improvements w.r.t. 2015 on Jinguanjia App

Challenges

1. Different from traditional task-oriented dialogues, customer-to-agent dialogues are usually very long, combined with task-oriented and chit-chat turns.
2. Easier for customer-to-agent dialogues to go viral with useful information scattered across multiple dialogue turns.
3. A new method, that can filter out pure chit-chat turns and reserve the most relevant task-oriented turns and chit-chat turns with marketing clues, is required.

Contributions

Our contributions in this paper are as follows:

- We propose a new industry scenario, where dialogues are long, combined with task-oriented and chit-chat turns.
- We design SHC to effectively select relevant dialogue history via slot-attention, and updates dialogue history memory.With SHC, our model could keep track of vital information and reduce the workload of dialogue state classifier.
- Our proposed LS-DST achieves state-of-the-art performance than other baselines on real insurance dialogue dataset.

Framework

LS-DST consists of three components:a dialogue encoder, a dialog memory manager, and a dialogue state classifier. We will introduce the model in detail in the following.

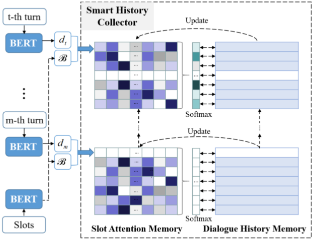

Figure 2: The Framework of smart history collector.

- Background
- Contributions

2 LS-DST

- Framework
- **Dialogue Encoder**
- Smart History Collector
- Dialogue State Classifier
- Objective function
- Experimental Objectives
- Datasets
- Performance
- Conclusion

Dialogue Encoder

- Before the encoder, we map customers' diverse oral expressions to standard ones.
- Then we use a pre-trained BERT as the context encoder.

For the pair of agent and customer utterance,$d_g^A = w_1^a, w_2^a, ..., w_{n_a}^a$,$d_g^C = w_1^c, w_2^c, ..., w_{n_c}^c$. A pre-trained BERT encode the word sequence:

$$d_t = BERT([CLS] \oplus d_t^A \oplus [SEP] \oplus d_t^C)$$

where $d_t \in \mathbb{R}^d$ is the output vector corresponding to $[CLS]$ token, d is the embedding size. Similarly, for the t-th turn, each domain-slot pair s (e.g., "insurance marketing-product name") and its value v_s^t are denoted as word sequences w^s and $w^{v_s^t}$, and let

$$s = BERT([CLS] \oplus w^s), y^{v_s^t} = BERT([CLS] \oplus w^{v_s^t})$$

s and $y^{v_s^t}$ are also output vectors corresponding to $[CLS]$ tokens.

Smart History Collector

Specifically, SHC maintains the memory of dialogue history and slot attention. The slot attention indicates how much the current dialogue relates to each domain-slot pair, which keeps updated at each turn. Dialogue history memory is the most relevant K historical turns selected with the help of slot attention. The framework of SHC is shown in Figure 3.

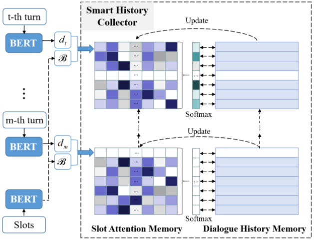

Figure 3: The Framework of smart history collector.

Smart History Collector

After dialogue encoder, we get dialogue context representation $D = [d_1, d_2, ..., d_{N_d}] \in \mathbb{R}^{d \times N_d}$, and initialize slot attention $A = [a_1, a_2, ..., a_{N_d}]$, where $a_t \in \mathbb{R}^J$, where:

$$a'_{jt} = s_j^T d_t,$$

$$a_{jt} = exp(a'_{jt}) / \sum_i exp(a'_{it})$$

For each $\mathcal{D}$, we use M^D and M^S to store dialogue history context embedding and its corresponding slot attention. Suppose both $M^D = [m_1^D, m_2^D, ..., m_K^D]$ and $M^S = [m_1^S, m_2^S, ..., m_K^S]$ have K columns (i.e., memory slots, in Figure 3, K=6).

Dialogue History Memory Pointer

As mentioned before, some turns of customer-to-agent dialogue are pure chit-chat. To filter these turns, we design a pointer, which is a simple classification task with score function defined as:

$$a_t = c^T \alpha_t$$

where $c \in \mathbb{R}^J$ is learnable parameter.

Dialogue History Memory Reading

If d_t contains useful information corresponding to domain-slot pair s_j, we read history memory by weighted summing dialogue history memories:

$$q_t^{s_j} = d_t + \sum_{k=1}^{K} [f_k^d(m_k^D, d_t) + f_K^s(m_k^D, s_j)] m_k^D$$

$$q_t^{s_j} = d_t + M^D w_t^{s_j}$$

where $f_k^d(\cdot,\cdot)$, $f_k^s(\cdot,\cdot)$ are related score functions, where

$$w_t^{s_j} = Softmax[(d_t^T M^D W_t^{s_j} + e_j^T M^S U_t^{s_j})^T] \in \mathbb{R}^K,$$

$W_t^{s_j}, U_t^{s_j} \in \mathbb{R}^{K \times k}$,$e_j$ is the j-th column of identity matrix I_J.

Dialogue History Memory Writing

If d_t contains vital information, we will update dialogue history memory, in this part,we adopt a simple first-in-first-out mechanism. When writing, the earliest dialogue turn m_1^D and its corresponding slot attention m_1^S would be replaced, and M^D be updated to $[m_2^D, m_3^D, ..., m_{K+1}^D]$, M^S be updated to $[m_2^S, m_3^S, ..., m_{K+1}^S]$, where $m_{K+1}^D = d_t$,

$$m_{K+1}^S = Softmax[a_t + \sum_{k=1}^{K} (d_t^T m_k^S) \cdot m_k^S].$$

Dialogue State Classifier

We use dialogue context representation $q_t^{s_j}$ as t-th dialogue state corresponding to s_j. Since output of BERT is normalized by layer normalization , $q_t^{s_j}$ is also fed into layer normalization:

$$\hat{y}_t^{s_j} = LayerNorm(q_t^{s_j}).$$

Objective function

$$p(v^t_{s_j} \mid d_t, s_j) = \frac{exp(-d(\hat{y}^{s_j}_t, y^{v^t_{s_j}}))}{\sum_{v' \in \mathcal{C}'_{s_j}} exp(-d(\hat{y}^{s_j}_t, y^{{v'}^t_{s_j}}))},$$

$$\mathcal{L}_{cls} = -\sum_{s_j \in \mathcal{B}'} logp(v^t_{s_j} \mid d_t, s_j),$$

$$\mathcal{L}_{ptr} = -\sum_{t=1}^{N_d} g_t log a_t + (1 - g_t) log(1 - a_t),$$

$$\mathcal{L}_{pred} = -\sum_{j=1}^{J} h^t_j log m^S_{jK} + (1 - h^t_j) log(1 - m^S_{jK}),$$

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{ptr} + \mathcal{L}_{pred}$$

Experimental Objectives

Extensive experiments are implemented to answer the following questions:

- RQ1: How does our proposed model perform compared with baselines ?
- RQ2: Can our designed SHC module deal with long and sparse dialogues ?

Datasets

- We collected a dialogue dataset (named LinsWOZ)from real insurance marketing scenario, which is annotated by five insurance experts according to user, agent states, and dialogue histories.
- To better understand the characteristics of LinsWOZ, we define Sparsity Degree to calculate a ratio of chi-chats over overall dialogue of each dialogue.
- At the same time, dialogues are grouped into five groups (2-5,5-10,10-15,15-20,>20) based on the length (total turns) of each dialogue.

Performance Comparison (RQ1)

To answer RQ1, we investigate the overall performance of our proposed model LS-DST and all baseline models.

Dataset	LinsWOZ	
Metric	Joint Acc.	Slot Acc.
TRADE	40.62	87.75
NADST	42.14	89.87
ML-BST	45.26	90.12
Bert-DST	47.08	91.26
SUMBT	**49.37**	**93.55**
LS-DST	57.13	96.78
LS-DST VS. best	15.72%	3.45%

Figure 4: Performance of our proposed LS-DST compared with baselines in LinsWOZ dataset. The results of the opti- mal baseline are expressed in bold.

The impact of SHC (RQ 2)

The dialogues in LinsWOZ dataset are grouped into five subsets of dialogues. Figure 5 shows the joint goal accuracy (Joint Acc.) and slot accuracy (Slot Acc.) with different length and different sparsity dialogue sub-sets.

Turns	2-5		6-10		11-15		16-20		>20	
Metrics	Joint Acc.	Slot Acc.	Joint Acc.	Slot Acc.	Joint Acc.	Slot Acc.	Joint Acc.	Slot Acc.	Joint Acc.	Slot Acc.
TRADE	47.13	91.01	46.66	90.66	44.48	88.91	42.07	87.82	39.20	87.23
NADST	48.62	93.50	47.54	91.95	45.03	90.48	44.88	89.34	40.65	89.55
ML-BST	50.87	95.29	49.91	93.20	47.16	92.09	46.35	91.06	44.24	89.27
Bert-DST	56.02	96.57	54.71	94.67	50.04	93.60	48.57	92.49	45.49	90.29
SUMBT	**59.41**	**97.12**	**56.90**	**95.84**	**54.17**	**94.69**	**52.71**	**94.05**	**47.25**	**93 .00**
LS-DST	64.64	98.08	62.27	97.16	60.38	97.02	59.06	96.98	55.71	96.63
LS-DST VS. best	8.80%	0.99%	9.44%	1.38%	11.46%	2.46%	12.05%	3.12%	17.90%	3.90%

Figure 5: Ablation study of SHC module in LinsWOZ dataset. The results of the optimal baseline are expressed in bold.

Conclusion

- we propose a new dialogue state tracking architecture containing three components:dialogue encoder, smart history collector and dialogue state classifier.
- SHC helps our model to keep track of the vital information and filter out pure chit-chat.
- We will consider to incorporate side information, such as domain specific knowledge, to improve our proposed LS-DST in future work.

Q&A

Thank you for your attention!

If you have any questions, please email to
magicyebi@163.com