Introduction to Towards End-to-End Reinforcement Learning of Dialogue Agents for Information Access

zvengin
zvengin@nii.ac.jp

December 18, 2017

1 General Introduction

Problem KB-InfoBot is a particular type of dialogue agent which helps users navigate a knowledge base in search of an entity without constructing complicated queries. Figure 1 shows an example.

Methods Given conversation history and user utterance, KB-InfoBot uses Neural Belief Tracker to estimate the probability distribution p_j^t and q_j^t (p_j^t is M multinomial distribution, $p_j^t(v), v \in V^j$ is the probability at turn t that the user constraint for slot j is v. q_j^t is a scalar probability of the user knowledge about the value of slot j, $q_j^t = Pr(\phi_j = 1)$ indicates the probability that user knows the vaule of slot j). Then Soft-KB Lookup is used to estimate the probability $P_\Gamma^t(i) = Pr(G = i|U_1^t)$. $P_\Gamma^t(i)$ indicates the probability that user is interested in row i of knowledge table, given utterance up to turn t. After that, the reuslt of Neural Belief Tracker and Soft-KB Lookup are summarized to internal state s^t. Finally, based on internal state s^t, we use Neural Policy Network to select an appropriate action π from action set. The frame of this system is shown in figure 2.

Techniques In this paper, author hopes to combine the result of Neural Belief Tracker (NBT) and the result of Soft-KB Lookup together by concatenating the results. However, the dimen-

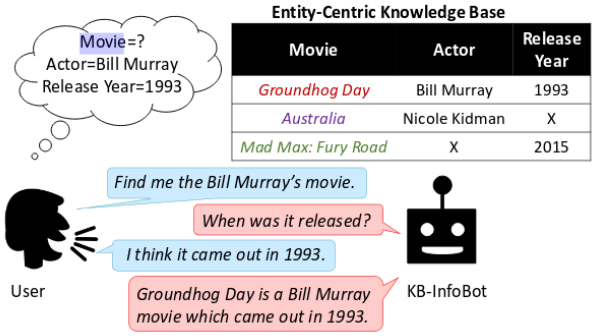

Figure 1: An example.

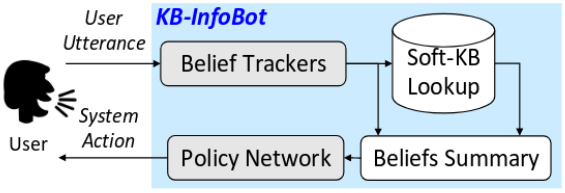

Figure 2: system frame

sion is too large to combine the results together. Therefore, author decreases the dimension by extracting summary statistic, that is author uses entropy to represent a probability distribution. $s^t = [p_1^t, p_2^t, ..., p_M^t, q_1^t, q_2^t, ..., q_M^t, P_\Gamma^t]$ is internal state and element p_j^t has $|V^j|$ dimensions. Therefore s^t has $\sum_j |V^j| + 2M + N$ dimensions. For each slot j, we compute $\hat{p}_j^t(v) = w_j^t(v) \propto \sum_{i:T_{i,j}=v} P_\Gamma^t(i) + p_j^0 \sum_{i:T_{i,j}=\phi} P_\Gamma^t(i)$ and then use the entropy of this new distribution $H(\hat{p}_j^t)$to represent the original distribution p_j^t. Finally the total dimension is decreased to $2M + 1$.

2 Details

2.1 Neural Belief Tracker(NBT)

The NBT is implemented by a GRU neural network. The input to this network is the utterance from beginning to turn t. The internal state $h_j^t \in R^d = GRU(x_1, ..., x_t)$ represents a summary of what the user has said about slot j till turn t, note $h_j^0 = 0$. The belief state can be expressed as following:

$$p_j^t = softmax(W_j^p h_j^t + b_j^p), \text{ where } W_j^p \in R^{V^j \times d}, b_j^p \in R^{V^j} \tag{1}$$

$$q_j^t = \delta(W_j^q h_j^t + b_j^q), \text{ where } W_j^q \in R^{1 \times d}, b_j^p \in R \tag{2}$$

2.2 Soft-KB Lookup

Soft-KB Lookup is used to determine the probability that the entity in which user is interested is row i of KB tabele, entity i.

$$P_\Gamma^t(i) = Pr(G = i|U_1^t) \propto \prod_j^M Pr(G_j = i|U_1^t) \tag{3}$$

where G means that the user's goal or intention, $G_j = i$ menans that with respect to the slot j, user is interested in row i. The prior distribution of G is uniform distribution $G \sim U[\{1, 2, ..., N\}]$.

$$
\begin{aligned}
&P(G_j = i|U_1^t) \\
&= \sum_0^1 Pr(G_j = i, \phi_j = \phi|U_1^t) \\
&= \sum_0^1 Pr(G_j = i|\phi_j = \phi, U_1^t) Pr(\phi_j = \phi|U_1^t) \\
&= q_j^t Pr(G_j = i|\phi_j = 1) + (1 - q_j^t) Pr(G_j = i|\phi_j = 0)
\end{aligned} \tag{4}
$$

$$
Pr(G_j = i|\phi_j = 0) = \frac{1}{N} \tag{5}
$$

where $\phi_j = 0$ means that user doesn't know the value of slot j

$$
Pr(G_j = i|\phi_j = 1) = \begin{cases} \frac{1}{N} & i \in M_j \\ \frac{p_j^t(v)}{N_j^v}(1 - \frac{|M_j|}{N}) & i \in M_j \end{cases} \tag{6}
$$

where M_j is a set which includes all rows i whose value of slot j is missing.

2.3 Neural Policy Network

Here author uses GRU neural network to construct a Neural Policy Network which converts internal state s^t to a probability distribution over the action set π.

$$
\begin{aligned}
h_\pi^t &= GRU(s^1, ..., s^t) \\
\pi &= softmax(W^\pi h_\pi^t + b^\pi)
\end{aligned} \tag{7}
$$

Refer the paper [1]

References

[1] Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. End-to-end reinforcement learning of dialogue agents for information access. *CoRR*, abs/1609.00777, 2016.