Advances in Educational Technology and Psychology, ISSN 2371-9400
2021 5th International Seminar on Teaching and Advances in Education
Vol. 25, 2021, pp. 110-113

Enhance Matching Model Performance by Deep Reinforcement Learning in Faq-Oriented Dialogue System

Xiaotong Pan[1,a,*], Zuopeng Liu[2,b]

[1]Xiaomi Ai, Beijing, China
[2]The Key Laboratory of Rich-Media Knowledge Organization and Service of Digital Publishing Content, Institute of Scientific and Technical Information of China, Beijing, China
[a] panxiaotong@xiaomi.com, [b] liuzuopeng@xiaomi.com
*corresponding author

ABSTRACT. in task-oriented dialogue system (TODS) we extract dialogue state against spoken language understanding findings, then calculate action distribution and determine one action as reply in policy learning, which is easy to define user feedback to adjust dialogue policy if we seem policy learning algorithm as an agent, dialogue state as slot filling result, task completion as reward, then train a deep reinforcement learning (DRL) model to determine which action would get more reward during prediction. Relatively in FAQ-oriented dialogue system (FODS) it is intractable to optimize question-answer policy out of user feedback since it is hard to define slots in FODS, all information is organized as frequently asked questions (FAQs), thus system action's selection in TODS' policy learning is converted to matching between user query and FAQ in FODS, that is hard to adjust matching algorithm by user feedback information directly. To tackle this issue, we propose an algorithm based on DRL that can improve matching performance by online user feedback, in contrast to previous matching approach which just made use of semantic similarity. By this method we increase interception ratio, decrease human transfer ratio and shorten average user interaction turn against various diverse business data in Xiaomi Intelligent Customer Service (XICS).

KEYWORDS: Dialogue system, Deep reinforcement learning

1. Introduction

In TODS we should defined slots and actions in advance, which represent what information should be filled by user during dialogue, the whole process is split into four modules, spoken language understanding (SLU), dialogue state tracking (DST), policy learning (PL) and natural language generation (NLG). In TODS partially observable Markov decision process (POMDP) is successfully applied in dialogue management for recent years[1-2].

Recent research focus on human-machine and machine-machine interaction by deep learning (DL) and reinforcement learning (RL) in dialogue system. With DRL[3] it is easy to optimize dialogue management policy by reward information. Value-based method tried to predict action's value according current state, selects best action by some policy, as well as DQN, DDQN, Duelling network or other methods. Policy-based method takes utility of policy gradient related algorithm or combination of value-based method and policy based method to adjust selection policy, as well as vanilla policy gradient, A2C, A3C, DDPG or other methods.

TODS' SLU process is the same with FODS, different with TODS there is neither slot nor action in FODS, frequently asked questions (FAQs) is a key element in FODS. We name one FAQ as Knowledge Item (KI) in XICS. Since it is hard to correlate user feedback with FODS' matching model together, in this paper we adjust answer display like ranking result and propose a new method by click information as user feedback to improve matching model's performance during Sorting phrase by DRL method. We introduce our algorithm in section 2 including Matching model, RL method and reward definition, experiments in section 3.

2. Algorithm

Our algorithm contain two components. Matching model calculate similarity between user query and KIs. RL reward controller collect user feedback as reward to fine-tune Matching model. The whole process is depicted in figure 1.

  DOI: 10.23977/ISTAE2021028

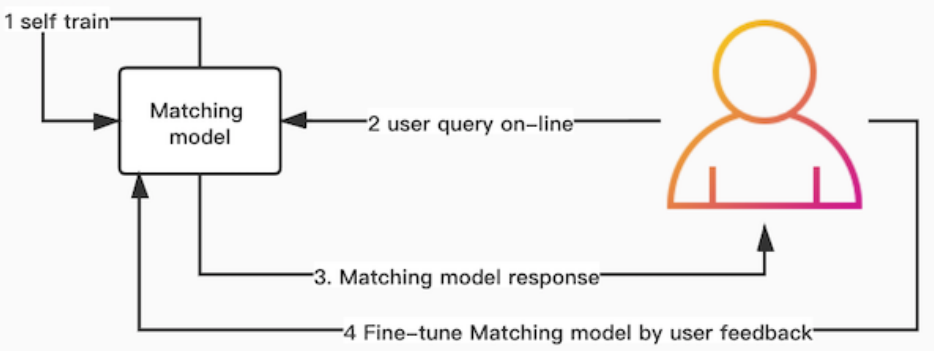

Fig.1 Architecture Overview

2.1 Matching Model

We use a Bi-LSTM model with cross-attention as experimental benchmark method, similar with BiMPM and ESIM containing context representation layer, matching layer, aggregation layer and prediction layer. Suppose user query as Q including n words, denoted as $q_0,\cdots,q_{n-1}$, i-th KI as p_i including m words denoted as $p_0^i,\cdots,p_{m-1}^i$. In context representation layer we use a Bi-LSTM layer and combine forward and backward features for j-th word in i-th KI, in matching layer we calculate attention between any word from query and any word from KI, in aggregation layer we leverage another Bi-LSTM layer with same hidden size to aggregate attention features together, in prediction layer we concatenate all information to calculate similarity between user query and KI.

The other enhanced model is BERT[4] based on pre-train model, which is used as feature extractor from [CLS] token, we fine-tune BERT model by a perceptron layer consuming [CLS] token's hidden state and calculate similarity.

2.2 Rl Method

We make experiments upon a model-based method DQN[5] and a hybrid method A2C. In DQN there are an evaluation mode and a target model, according to a quadruple tuple $\langle s_t, a_t, r_t, s_{t+1}\rangle$ in time t evaluation model calculate Q-value given s_t and a_t with s_{t+1} target model extract maximum Q-value according to the responding action a_t , loss function is as formula (1), finally we use evaluation network's result during prediction.

$$L(\theta_{eval}) = \mathrm{E}_{(s_t,a_t,r_t,s_{t+1})\sim U(D)}(r_t + \gamma \max_a Q(s_{t+1}, a; \theta_{tar}) - Q(s_t, a_t; Q_{eval}))^2 \quad (1)$$

A2C algorithm leverage another network whose top layer is all actions' Softmax distribution with a different triplet tuple $\langle s_t, a_t, r_t\rangle$, r_t is multiplied with cross-entropy as final loss function. A2C algorithm would increase good action's probability and decrease bad action's probability.

2.3 Reward Definition & Advantage Function

How to define reward is the most necessary factor when using RL to improve performance continuously. QA board is designed in XICS as figure 2. XICS would recommend K relevant KIs as candidates besides direct answer, thus whether to click against recommendation KI could seemed as a reward in RL algorithm.

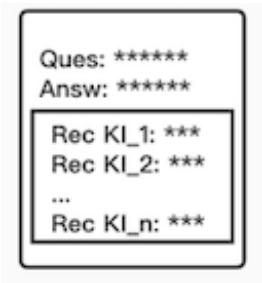

Fig.2 : Xics Qa Board

By using user behaviour we design two rules as below:

a. User doesn't click any recommendation KI, ask a new question whose intent is different with previous question. In this case we get a sample $\langle s_t, a_t, 1, s_{t+1} \rangle$ whose s_t is extracted from user's previous question, a_t is extracted from previous reply KI, s_{t+1} is extracted from new question. Meanwhile we get K $\langle s_t, a_t, -1, s_{t+1} \rangle$ samples whose s_t and s_{t+1} are the same with $\langle s_t, a_t, 1, s_{t+1} \rangle$ sample except that a_t is extracted from each recommendation KI.

b. User click any recommendation KI. In this case we get a sample $\langle s_t, a_t, 1, s_{t+1} \rangle$ whose s_t is extracted from current question, a_t is extracted from clicked recommendation KI, s_{t+1} is extracted from user's next question, if there is no next question, then s_{t+1} is set to null, we remove $\gamma \max_a Q(s_{t+1}, a; \theta_{tar})$ part of loss function in formula (1). We don't extract $\langle s_t, a_t, -1, s_{t+1} \rangle$ from reply KI since user may be curious about a relevant KI in recommendation lists.

3. Experiments

3.1 Dataset

Our traffic is more than 1 million episodes from different scenarios every day in XICS' online service, at this point we have more than 50 customer service's scenarios including Chinese language and English language. From QA log we extract $\langle s_t, a_t, 1, s_{t+1} \rangle / \langle s_t, a_t, -1, s_{t+1} \rangle$ by two rules defined in section 2.3. We just update models by offline method other than online method since it is hard to converge with less training data in online learning.

3.2 Result

We train two Matching models benchmark model and BERT-base model respectively, then leverage model-based DQN method and A2C method to fine-tune Matching model. In three scenarios with different KI size, we do experiments to compare Precision and F1-score, the result is shown in Table 1 and Table 2. From the table we observe that Bert base model with A2C method perform best. We speculate that BERT model extract better features against Bi-LSTM model, meanwhile A2C method collects more data than DQN thus model free method performed better than model based method. Comparison between scenario with 2,000 KIs and 300 KIs, we find that the change isn't obvious because in scenario with 300 KIs there is less training data than in scenario with 2,000 KIs. Comparison between scenario with 10,000 KIs and 2,000 KIs, we find that scenario with more KIs perform worse than less KIs because more KIs correspond with more action space, which is harder to select.

Table 1 : Experimental Performance in Precision Metric

	10,000 KIs	2,000 KIs	300 KIs
Bi-LSTM + DQN	0.760	0.781	0.785
Bi-LSTM + A2C	0.793	0.791	0.792
Bert-base + DQN	0.822	0.836	0.836
Bert-base + A2C	0.841	0.852	0.853

Table 2 : Experimental Performance in F1-Score Metric

	10,000 KIs	2,000 KIs	300 KIs
Bi-LSTM + DQN	0.603	0.622	0.620
Bi-LSTM + PG	0.638	0.640	0.649
Bert-base + DQN	0.638	0.657	0.659
Bert-base + A2C	0.642	0.669	0.667

4. Conclusion

In this paper we propose an algorithm with DRL in FODS. With RL policy FODS perform better in matching task. It prove that integration RL with FODS is a good approach to improve dialogue system's matching performance. In the future we will try more experiments upon human transfer and solved/unsolved feedback information, and some user questions with emotion can give feedback information, too.

Acknowledgement

Teer: This work was supported by the Fund of the key laboratory of rich-media knowledge organization and service of digital publishing content (ZD2019-10/03)

References

[1] Roy Nicholas, Pineau Joelle, Thrun Sebastian."Spoken dialogue management using probabilistic reasoning", In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Hong Kong, China, pp. 93-100, 2000.

[2] Jason Williams, Steve Young. Partially observable Markov decision processes for spoken dialog systems. Computer Speech and Language, vol.21, no. 2, pp. 393–422, 2007.

[3] Mnih, Volodymyr, Kavukcuoglu, Koray et al. Martin. Playing atari with deep reinforcement learning. In NIPS Deep Learning Workshop, 2013.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[5] Matthew Hausknecht, Peter Stone. Deep recurrent q-learning for partially observable mdps. arXiv preprint arXiv:1507.06527, 2015.