

End-to-end Dialog Systems with Numerical Slot Filling

Hongjie Shi

Megagon Labs, Tokyo, Japan, Recruit Co., Ltd.

shi.hongjie@megagon.ai

1 Introduction

Task-oriented dialog systems which assist users to complete tasks like hotel reservation, are drawing great attentions among both research and industry. Compared to conventional pipelined system, recently emerging end-to-end trainable dialog systems are showing many favorable characteristics — because of the neural models that directly learn from chatlogs of human-to-human conversation employed, such systems hold the promise of low data preparation cost, flexible response generation and the ability to evolve with new data.

In this work, we are going to explore the possibility to bring this end-to-end trainable framework to the hotel reservation chatbot application, where we encounter two new problems: 1. numerical slots-filling and 2. multi-turn dialog management. To the best of our knowledge, both of them can not be fully solved using currently available end-to-end frameworks. In this paper, we will focus on these two problems and propose possible workarounds which can lead to satisfactory results.

2 Problem description

The hotel reservation application requires a dialog system to fill three slots with integer — number of adults (`slot:num_adult`), number of primary school children (aged 6-12) (`slot:num_c6_12`), and number of preschool children (aged 0-5) (`slot:num_c0_5`). These numerical slots are necessary because the applicable room plan (number of beds, quantity of amenities) and the pricing (food cost etc.) vary on the number of adults and children. This numerical slot-filling problem is also widely applicable to other domains such as restaurant reservation or flight booking, with slightly different slot configurations.

The challenges of building such dialog system

mainly lie in two aspects. First challenge is the difficulty in the numerical slot value inference. Unlike most task-oriented dialog systems or datasets such as (Wen et al., 2016; Henderson et al., 2013), where the slot filling can be either solved as a named entity extraction problem or a multi-label classification problem, the numerical slot-filling requires additional reasoning and calculation. For examples the simple expression “*My wife and me*” means 2 adults, and “*4 including 1 baby*” implies 3 adults. And moreover, the numerical inference sometimes involves with multi-turn dialog context, which brings to the second challenge.

Second challenge is the multi-turn dialog management. Many previous task-oriented dialog systems are designed in a turn-wise manner — the systems ask the question for particular slot in each turn and expect user to give explicit answer within that turn. If no specific slot value can be extracted from the response, the system will simply repeat the same question. This behavior is unfavorable for the numerical slot-filling, because of the likely ambiguity in the user responses. For example, no target slot value can be determined from the user response “*4 people including 2 kids*”, while human agent may ask drill-down questions such as “*How old are the children*” to address this ambiguity. To achieve this human-level conversation, a dialog system capable of managing multi-turn strategy such as ask drill-down questions, is desirable.

3 Present methods

Several end-to-end model architectures have been proposed for task-oriented dialog system. Wen et al. (2016) proposed a modularly connected neural networks to enable end-to-end training. Later Lei et al. (2018) simplified this architecture to a single sequence-to-sequence model (SEQUICITY), which not only reduced the training cost but also improved the performance. More recently more

advanced model like HaGAN has been applied to end-to-end learning (Fang et al., 2019). Wu et al. (2019) also explored the possibility of applying recent large pre-trained language model such as BERT and GPT-2 to the task-oriented dialog system.

After survey and review on different models, we consider the SQUICITY framework a particularly good point to start because of its simplicity and extendability. The key idea is to encode the dialog states (slot values) into a text format which can be concatenated to the utterances, so that any seq-to-seq models can handle both slot-filling and language generation at the same time. In this way, the model complexity and the training procedure are simplified (refer to original paper for more details). Recently published T5 model (Raffel et al., 2019) also demonstrates the promising performance and the wide applicability of such text-to-text format training. We consider this SQUICITY framework using seq-to-seq model has great potential, and therefore in this paper we chose this framework as our base model.

4 Proposed methods

4.1 Slot-filling with numerical reasoning

In order to enable the seq-to-seq model to perform numerical reasoning, we train the model to predict arithmetic expressions instead of numeric values. For example for the utterance “*three men and two women*”, we modify the target output to be ‘3+2’ instead of the numeric value ‘5’ during training. This encourages the seq-to-seq model to simply copy values from the input sentence, rather than manipulate the number directly. This method is also inspired by recent state-of-the-art models from the Discrete Reasoning Over Passages (DROP) dataset (Dua et al., 2019), where most models are trained to predict numerical spans and math operations respectively (Ran et al., 2019; Andor et al., 2019). Intuitively, by doing so, we can achieve better generalization performance because it can easily handle unseen combinations of different numbers and math operations.

4.2 Multi-turn dialog management for ambiguous responses

The original SQUICITY model only takes one single turn of previous utterance and slot values as model input for the response generation. This mechanism reduces the training cost, however,

may hinder the model from learning multi-turn dialog strategy. For example in the following dialog:

Agent: How many people is the reservation for?
User: Four people including two kids. (1) (total_num:4 num_child:2)
Agent: ...
User: ...
Agent: How old are the two children? (2)
User: One is 5 and the other is 8. (num_adult:2 num_c6_12:1 num_c0_5:1)

It will not be possible for system to ask questions like (2) without being aware of earlier user utterance (1). To address this problem, we use additional slots to track down all necessary dialog context. We call them *context slots*. In this particular example, we use two context slots — total_num slot with value of 4 and num_child slot with value of 2 to track the information mentioned in user utterance (1). We treat these context slots just like other numerical slots — they will be carried on to the next turn’s input until the goal is achieved, so that the model can refer to them at any position of the dialog. With the help of context slots, the dialog system can generate context-aware questions with less effort, and also is able to learn multi-turn dialog strategy from less data.

5 Experiment and results

5.1 Slot-filling with numerical reasoning

To collect training and evaluation dataset with arithmetic expression, crowd sourcing service was used. We ask crowd workers to compose utterance using numbers given in the instruction, while avoiding directly including the numeric answer in the sentence, so that each collected utterance requires numerical reasoning for inference. The collected data follows the 5 patterns listed below:

Slot value	Description / Examples
num_adult n	mentioning n number of adults E.g. <i>One adult (1), My wife and me (2)</i>
num_adult $n_1 + n_2$	mentioning n_1, n_2 numbers of adults E.g. <i>One adult and one middle school child. (1+1)</i>
num_adult $n_1 + n_2 + n_3$	mentioning n_1, n_2, n_3 numbers of adults E.g. <i>Two of us, two friends and one colleague. (2+2+1)</i>
num_adult $n_1 - n_2$	mentioning n_1 number of people in total and n_2 number of primary school child or preschool child. E.g. <i>4 people including one preschool child. (4-1)</i>
num_adult $n_1 - n_2 - n_3$	mentioning n_1 number of people in total and n_2, n_3 number of primary school child and preschool child respectively. E.g. <i>4 people including one 8-year-old child and one 3-year-old child. (4-1-1)</i>

All utterances are trained with the arithmetic expressions as shown inside the brackets above. We also compared the proposed method to training the model with the execution results of algorithmic ex-

pressions. The result is summarized in the table below:

training data #	Predicting numerical values F1	Predicting arithmetic expressions F1
574	0.37	0.91
1148	0.80	0.94
2296	0.93	0.94

Our result shows that the proposed method (predicting algorithmic expression) outperforms predicting numeric value by a huge margin when the training data size is small. However, by increasing the training data size, the performance gap between two methods can be greatly reduced. These results can be interpreted as the following two reasons. 1. Algorithmic expression prediction have superior generalization performance for small size of training data, because it can easily handle unseen combination of numbers. On the other hand predicting with numeric value requires the model to also learn to manipulate numbers directly, therefore it may need more instances to train. 2. The algorithmic expressions appeared in the dataset is quite simple with limited range and variations. It is possible to train a seq-to-seq model with pretrained word embedding to be able to do simple calculation. This observation is consist with one recent paper, which also reported the good performance of neural models on addition calculation within the training range (Wallace et al., 2019).

In the real application, we can combine these two models to furthermore boost the performance in a ensemble learning way. And also, when two models give completely different answers, we can also tune the dialog system to confirm with user.

5.2 Multi-turn dialog management for ambiguous responses

To achieve dialog management resemble to real human-human conversation, we collected around 900 hotel reservation dialogs from pairs of workers who played agent or user roles. Each dialog covers all topics in hotel reservation, including location, price range, preference and so on. We then analyzed all sub dialog segments concerning `total_num/num_child` slot from each dialog, and extracted 7 representative drill-down questions as listed below:

- *Are all people above middle school students?*
皆様中学生以上の大人でしょうか？
- *Are there any children in the group?*
お子様はいらっしゃいますか？

- *Are there any children who are primary school students or below?*
小学生以下のお子様はいらっしゃいますか？
- *Are all people adults?*
皆様大人の方でいらっしゃいますか？
- *How old is the child?*
お子様のご年齢をお伺いできますか？
- *If there is any child in the group, could you please tell me their ages?*
お子様がいらっしゃる場合は年齢を教えてくださいませんか？
- *Is it <total_num> adults?*
大人<total_num>名様でよろしいでしょうか？

In order to collect more variations of possible user utterances which are applicable to these questions, again we used crowd sourcing service and asked workers to fill in the blank of the dialog below:

Agent: How many people is the reservation for?
User: (a)
Agent: <one of the questions shown in above list>
User: (b)
Agent: Alright, so it is <n1> adults, <n2> child (6-12) and <n3> child (0-5).

Example of collected dialogs:

Agent: How many people is the reservation for? お泊りの人数はお決まりでしょうか？
User: For 3 people. 子供を入れて全部で6名です。
Agent: How old are the children? お子様のご年齢をお伺いできますか？
User: We have one primary school child, two preschool children. 小学生が1名、幼児が2名います。
Agent: Alright, so it is 3 adults, 1 child (6-12) and 2 children (0-5). かしこまりました。大人3名、小学生2名、幼児1名ですね。

With these data, our model is ready to train for the multi-turn dialog strategy. During the training, turn (a) is trained with context slots (Sect. 4.2) and next-turn agent question. Turn (b) is trained with target slot values consistent with the last agent utterance, and a special token <DIALOG_END>.

To evaluate the model, we extracted 20 dialog segments that contain drill-down questions from actual human-human dialogs as hold-out test dataset, and only train the model with 200 crowd sourcing dialogs. Samples of generated responses can be found in table 1. Human evaluation shows that 80% of model generated responses are reasonable, however compared to actual human dialogs, the responses generated by model tend to be less diverse. In particular the question which quotes user context — “*Is it <total_num> adults?*”, never appears in the model output, even though

Agent utterances	User utterances	Predicted slots	Generated response
How many people is the reservation for? お泊りの人数はお決まりでしょうか？ Are there any children in the family? ご家族の中にお子様はいらっしゃいますか？	Four members of my family. 家族4名です。 No there isn't. いません。	total_num 4 num_adult 4 num_c6_12 0 num_c0_5 0	Are there any children in the group? お子様はいらっしゃいますか？ <DIALOG_END>
How many people is the reservation for? お泊りの人数はお決まりでしょうか？ Could you tell me the age of your child? お子様のご年齢を教えてくださいませんか？	2 adults and 1 child please. 大人2人子供1人でお願います。 Four years old! 4歳です！	num_adult 2 num_child 1 num_adult 2 num_c6_12 0 num_c0_5 1	How old is the child? お子様のご年齢をお伺いできますか？ <DIALOG_END>
How many people is the reservation for? お泊りの人数はお決まりでしょうか？ Are all people adults? 皆様大人の方でいらっしゃいますか？	For 3 people. 3名です。 2 adults and 1 child. 大人2名と子供1名です。	total_num 3 num_adult 2 num_c6_12 0 num_c0_5 1	Are all people adults? 皆様大人の方でいらっしゃいますか？ <DIALOG_END>
How old is the child? お子様のご年齢をお伺いできますか？	10 years old. 10歳です。	N/A	N/A

Table 1: Samples of test dialogs with model outputs, including the slot values and next response. The model is trained and predicted in Japanese. Red bold texts are the results considered to be incorrect.

it has same number of training data as the other questions. This is also a common issue that has been studied in previous general-purpose and task-oriented dialog models (Shao et al., 2017; Rajendran et al., 2018). And in our case, the robustness to unseen dialog flows (e.g. 3-turn drill-down as the last example shown in the table 1) is also an issue to be improved in the future.

6 Conclusion and future work

In this paper, we proposed two methods for improving the original end-to-end dialog system on numerical slot-filling. By training the model to predict arithmetic expressions, the dialog system can handle numeric reasoning more robustly, and with newly designed context slots, the dialog system is able to generate multi-turn questions for ambiguous user responses.

Future work may include extending the current seq-to-seq network to more recent large-scale pre-trained models such as RoBERTa, as suggested in (Talmor et al., 2019), for a better performance in reasoning task. And also the proposed multi-turn dialog management approach should be extensively tested on other slots and domains.

Acknowledgments

I would like to thank Hidekazu Tamaki for the help of data collection, and Prof. Yuki Arase for the helpful research advice.

References

Wen, Tsung-Hsien, et al. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.

Henderson, Matthew, et al. 2013. Dialog state tracking challenge 2 & 3.

Lei, Wenqiang, et al. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. *Proceedings of ACL 2018*.

Fang, Ting, et al. 2019. HaGAN: Hierarchical Attentive Adversarial Learning for Task-Oriented Dialogue System. *International Conference on Neural Information Processing*.

Wu, Qingyang, et al. 2019. Alternating Recurrent Dialog Model with Large-scale Pre-trained Language Models. *arXiv preprint arXiv:1910.03756*.

Raffel, Colin, et al. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Dua, Dheeru, et al. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. *arXiv preprint arXiv:1903.00161*.

Ran, Qiu, et al. 2019. NumNet: Machine Reading Comprehension with Numerical Reasoning. *arXiv preprint arXiv:1910.06701*.

Andor, Daniel, et al. 2019. Giving BERT a Calculator: Finding Operations and Arguments with Reading Comprehension. *arXiv preprint arXiv:1909.00109*.

Wallace, Eric, et al. 2019. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. *arXiv preprint arXiv:1909.07940*.

Talmor, Alon, et al. 2019. oLMpics—On what Language Model Pre-training Captures. *arXiv preprint arXiv:1912.13283*.

Shao, Louis, et al. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. *arXiv preprint arXiv:1701.03185*.

Rajendran, Janarthanan, et al. 2018. Learning end-to-end goal-oriented dialog with multiple answers. *arXiv preprint arXiv:1808.09996*.