Multi-domain Dialogue State Tracking via Pre-trained Context Model

Rongwen Zhao, Zekun Zhao, Zhengqi Liu
University of California, Santa Cruz
{rzhao17, zzhao99, zliu194}@ucsc.edu

Abstract

Multi-domain Dialogue State Tracking, which predicts user goals and requests at every dialogue turn, is one of the key components in task-oriented dialog systems. Plenty of approaches based on end-to-end networks have been proposed to solve this problem and demonstrated state-of-the-art performance. In this project, we utilize a pre-trained BERT based encoder-decoder model to make predictions given multi-domain context. To illustrate the effectiveness of the model on multi-domain dialogue, we experiment with the MultiWOZ-2.1, a multi-domain task-oriented dialogue dataset. Our experiments show that the model is able to scale to multi-domain applications, though the performance is not as good as the state-of-the-art. Also, we discuss some possible reasons accounting for the poor performance.

1 Task Definition

Dialogue state tracking (DST) (Rastogi et al., 2017) is the core part of a dialog system (DS)[1]. People also use a dialog manager (DM) to describe this component in a dialog system. It is responsible for the state and flow of the conversation. Generally, people have specific goals at their every dialogue turn, which is even more clear in a task oriented dialogue. Tracking important information in a dialogue can helps communication more efficient, such as restaurant reservation or ticket booking. We define the goal of DST module is to extract user goals during conversation and encode them as a compact set of dialogue states.

2 Motivation

In a Dialogue system what we actually get from the user when he uses our system is either speech or text. If it is speech, we can run it through ASR and get the text(utterance). The first thing you need to do when you get the utterance from the user, is to understand what does the user want, and this is the intent classification problem. We usually think of intent as actually a form that a user needs to fill in, And each intent has a set of fields or so-called slots that must be filled in to execute the user request. And we need a slot tagger to extract slots from the user utterance. Thinking of it as a sequence tagging and we can solve it as by sequence tagging tasks. Also, we need to add or update context to our intent classifier and slot tagger. Context is actually some information about what happened previously. If we focus on the Dialogue State Tracking Part, it looks like Figure 1.

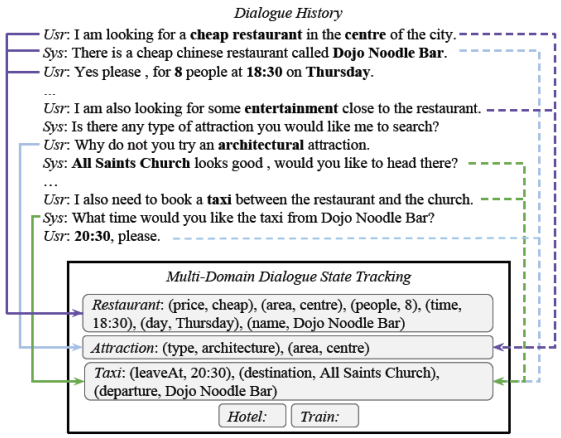

Figure 1: A typical Multi-Domain Dialogue State Tracking system.

3 Related Work

Traditional approaches use a Spoken Language Understanding (SLU) unit that utilizes a semantic dictionary to hold all the key terms, rephrasings and alternative mentions of a belief state. The SLU then delexicalises each turn using this semantic dictionary, before it passes it to the BT component. . However, this approach is not scalable to multi-domain dialogues because of the effort required to

[1] https://en.wikipedia.org/wiki/Dialogue_system

define a semantic dictionary for each domain.

Many Traditional approaches in this field are not scalable to multi-domain dialogues because of the effort required to define a semantic dictionary for each domain.

3.1 Multi-Domain Belief Tracking

The Neural Belief Tracker (NBT) (Ramadan et al., 2018), use word embeddings to alleviate the need for delexicalisation and combine the SLU and BT into one unit, mapping directly from turns to belief states. However, the NBT model does not tackle the problem of mixing different domains in a conversation. Moreover, as each slot is trained independently without sharing information between different slots, scaling such approaches to large multi-domain systems is greatly hindered. Domain tracking is considered as a separate task but is learned jointly with the belief state tracking of the slots and values.

The core idea in Multi-Domain Belief Tracking is to leverage semantic similarities between the utterances and ontology terms to compute the belief state distribution. In this way, the model parameters only learn to model the interactions between turn utterances and ontology terms in the semantic space, rather than the mapping from utterances to states. Consequently, information is shared between both slots and across domains. Additionally, the number of parameters does not increase with the ontology size.

3.2 TRADE

TRAnsferable **D**ialogue stat**E** generator (**TRADE**) (Wu et al., 2019) is a model that instead of predicting the probability of every predefined ontology term, but directly generates slot values. So, TRADE can directly track those slots that are present in a new domain. Trade reaches a joint accuracy of 48.62% and slot accuracy of 96.92% working on MultiWOZ_2.1. As figure 2 shows, the architecture of TRADE could be divided into three parts, the utterance encoder, the state generator, and the slot gate. The utterance encodes simply encode the input utterances into vectors. The state generator will decode J times independently for all the possible (domain, slot) pairs. At the first decoding step, the state generator will take the j-th (domain, slot) embeddings as input to generate its corresponding slot values and slot gate. The slot gate predicts whether the j-th (domain, slot) pair is triggered by the dialogue.

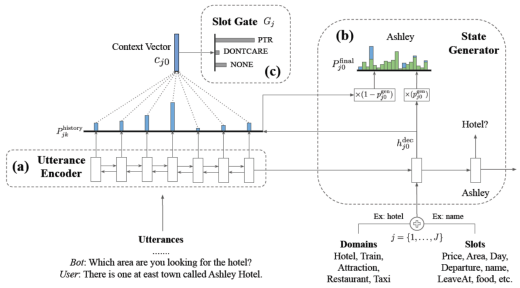

Figure 2: Architecture of the TRADE model.

3.3 SUMBT

Slot-**U**tterance **M**atching **B**elief **T**racker (**SUMBT**) (Lee et al., 2019) is a model that learns the relations between domain-slot-types and slot values appearing in utterances through attention mechanisms based on contextual semantic vectors. It reaches a joint accuracy of 46.649%, and a slot accuracy of 96.44% on the Multi_WOZ Corpus.

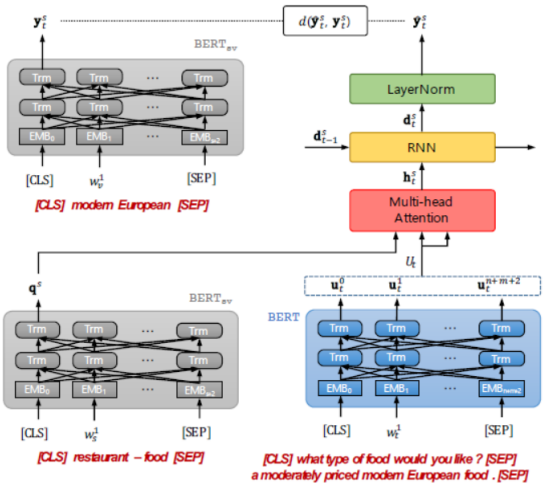

Figure 3: Architecture of the SUMBT model.

Figure 3 is the architecture of the SUMBT model. From the figure, we can see that the model uses three BERT encoder (the grey box and the blue box) to encode three different inputs. In the figure, ”modern European” is the value, ”restaurant - food” is the domain - slot pair, and the sentence in the blue box is the utterance. The red box is a slot-utterance matching network, the orange box is a belief tracker, and the green box is a non-parametric discriminator.

4 Approach

In this section, we will introduce the model for this project. Our model is based on the previous work

ComerNet (Ren et al., 2019). As shown in Figure 4, the model consists of three independent encoders and three stacked decoders.

4.1 Encoder Module

Each encoder contains a pre-trained bert embedding layer and two bidirectional LSTM layers. For each encoder, it will receive the user utterance, the system utterance and previous state at the current turn. Then it will generate the hidden representation for all of the inputs. First, each sequence of words, like user utterance or system utterance, will be sent into a pre-trained bert model and the corresponding contextual embedding will be obtained. Second, since for some slots, it may have more than one word. In order to deal with such cases, we can feed the word vectors into the bert and take the average of the word vectors as the extra slot embedding. A static vocabulary embedding is also constructed by feeding each word in the bert vocabulary into the BERT model. In this way, we concatenate the extra slot embedding and static vocabulary embedding together to get the final static word embedding for the previous state. When the contextual embedding of each user utterance, system utterance and the static embedding of the previous state are obtained, each of three embeddings will be fed into a two-layer bidirectional LSTM. The parameters are shared across all of these bidirectional LSTMs.

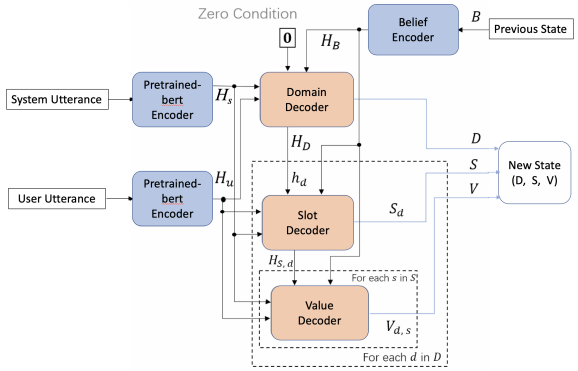

Figure 4: The architecture of the proposed model.

4.2 Decoder Module

Decoder takes the same representation inputs from the user utterance encoder, system utterance encoder and previous state encoder, as shown in Figure 5. Then it will generate corresponding sequence hierarchically. These three decoders share same parameters. For the first decoder, except the previous three embeddings, it also takes in a zero vector as its initial state and generate a sequence of domains D, as long with the hidden representation of domains H_D. For the second decoder, it will take h_d from previous representation H_D, where d is a domain in D. Then it will generate a slot sequence S_d and its corresponding representation $H_{S,d}$. Last, the third decoder will output the value sequence $V_{d,s}$ given the corresponding representation $h_{s,d}$ for each slot s in S. After this process, we can update the state with the new pair (d, (s_d, $V_{d,s}$)) and continue this process until a dialogue is finished.

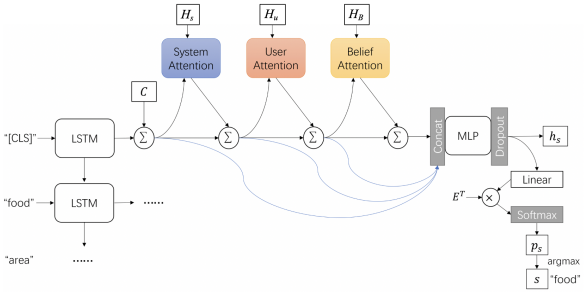

Figure 5: The architecture of the decoder module.

5 Experiments

5.1 Data

The dataset we use is the **MultiWOZ_2.1(Eric et al., 2019)**, which is a fully-labeled collection of human-human written conversations including more than 10k dialogues. Each dialogue consists of goal, multiple user, system utterances and a belief state. It is a multi-domain dataset that have 7 different domains. They are 'Booking', 'Restaurant', 'Hotel', 'Attraction', 'Taxi', 'Train' and 'General'.

5.2 Evaluation Measures

The metrics we use in our experiment are joint goal accuracy and slot accuracy(Wu et al., 2019). The joint goal accuracy compares the predicted dialogue states to the ground truth B_t at each dialogue turn t, and the output is considered correct if and only if all the predicted values exactly match the ground truth values in B_t. The slot accuracy, on the other hand, individually compares each (domain, slot, value) triplet to its ground truth label.

5.3 Settings

The DistilBERT model is used for all of the token representations, including contextual and static embedding, since it will save much time compared to other BERT models. The embdding size is 768,

since a DistilBERT model contains 768 layers. For the training stage, we adapt the ADAM optimizer with the learning rate of 5e-5 for optimization and the CrossEntropy loss for the evaluation with a batch size of 64. We also utilize some useful training techniques, like dropout and weight decay. Besides, all of the weights are initilized by Kaiming initialization (He et al., 2015).

6 Results and Analysis

In order to evaluate the performance of our model, we compare the results generated by our model with several state-of-the-art models, as shown in Table 1. We got 12.48% for the joint accuracy and 69.34% for the slot accuracy. As we can see, the result of our model is not very good compared to others, especially as for the joint accuracy. There are some possible reasons to account for bad performance. First, we only trained one epoch for each experiment of our model. Since it consumes several hours to train one epoch for the whole data. Hence, if we can train more epochs, the performance will be get improved. Second, the performance of joint accuracy is even worst than slot accuracy, which means the representation between multiple domains is not appropriate. We need to come up with better representations for user utterance, system utterance and previous state at the current turn.

7 Challenges and Future Work

During this project, we encounter some challenging issues. First, the dataset we used is profoundly complex since multiple domains are involved in one dialogue simultaneously. Second, to represent involved domains and related slot/value pairs well are also hard. For the future work, we could incorporate pre-defined ontologies consisting of a set of all possible slot types and values into our model. Another idea is maybe we could also model multi-domain DST as a question answering task after getting representations from pre-trained models.

8 Teamwork

All of our team members have contributed lots of efforts for this project from beginning to end. We have weekly meeting for discussing current progress with difficulties and the further plan. Everyone has participated in the project presentation and report writing. Rongwen Zhao mainly focused on conducting experiments. Zekun zhao paid more attention on the part of evaluation of related works for the new dataset, and comparison of performance from different models. Zhengqi Liu was responsible for analyzing dataset and related works.

Model	Joint	Slot
NBT (Ramadan et al., 2018)	15.57	89.53
TRADE (Wu et al., 2019)	45.60	-
DST-Picklist (Zhang et al., 2019)	53.30	-
DSTQA (Zhou and Small, 2019)	51.17	97.21
Our	12.48	69.34

Table 1: The comparison of different model results.

References

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. Sumbt: Slot-utterance matching for universal and scalable belief tracking. *arXiv preprint arXiv:1907.07421*.

Osman Ramadan, Paweł Budzianowski, and Milica Gašić. 2018. Large-scale multi-domain belief tracking with knowledge sharing. *arXiv preprint arXiv:1807.06517*.

Abhinav Rastogi, Dilek Hakkani-Tür, and Larry Heck. 2017. Scalable multi-domain dialogue state tracking. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 561–568. IEEE.

Liliang Ren, Jianmo Ni, and Julian McAuley. 2019. Scalable and accurate dialogue state tracking via hierarchical sequence generation. *arXiv preprint arXiv:1909.00754*.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. *arXiv preprint arXiv:1905.08743*.

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. 2019. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*.

Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *arXiv preprint arXiv:1911.06192*.