Towards Universal Dialogue State Tracking

Liliang Ren, Kaige Xie, Lu Chen and Kai Yu
Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Eng.
SpeechLab, Department of Computer Science and Engineering
Brain Science and Technology Research Center
Shanghai Jiao Tong University, Shanghai, China
{renll204,lightyear0117,chenlusz,kai.yu}@sjtu.edu.cn

Abstract

Dialogue state tracking is the core part of a spoken dialogue system. It estimates the beliefs of possible user's goals at every dialogue turn. However, for most current approaches, it's difficult to scale to large dialogue domains. They have one or more of following limitations: (a) Some models don't work in the situation where slot values in ontology changes dynamically; (b) The number of model parameters is proportional to the number of slots; (c) Some models extract features based on hand-crafted lexicons. To tackle these challenges, we propose StateNet, a *universal* dialogue state tracker. It is independent of the number of values, shares parameters across all slots, and uses pre-trained word vectors instead of explicit semantic dictionaries. Our experiments on two datasets show that our approach not only overcomes the limitations, but also significantly outperforms the performance of state-of-the-art approaches.

1 Introduction

A task-oriented spoken dialogue system (SDS) is a system that can continuously interact with a human to accomplish a predefined task through speech. It usually consists of three modules: input, output, and control. The control module is also referred to as *dialogue management* (Young et al., 2010; Yu et al., 2014). It has two missions: dialogue state tracking (DST) and decision making. At each dialogue turn, a state tracker maintains the internal state of the system based on the information received from the input module. Then a machine action is chosen based on the dialogue state according to a dialogue policy to direct the dialogue (Chen et al., 2018).

The dialogue state is an encoding of the machine's understanding of the whole conversation. Traditionally, it is usually factorized into three distinct components (Young et al., 2013): the user's *goal*, the user's action, and the dialogue history. Among them, the user's goal is most important, which is often simply represented by *slot-value* pairs. In this paper, we focus on the tracking of the user's goal.

Recently, the dialogue state tracking challenges (DSTCs) (Williams et al., 2013; Henderson et al., 2014a,d) are organized to provide shared tasks for comparing DST algorithms. A various of models are proposed, e.g. rule-based models (Wang and Lemon, 2013; Sun et al., 2014a; Yu et al., 2015, 2016; Sun et al., 2016b), generative statistical models (Thomson and Young, 2010; Young et al., 2010, 2013), and discriminative statistical models (Lee and Eskenazi, 2013; Lee, 2013; Sun et al., 2014b; Xie et al., 2015; Sun et al., 2016a; Xie et al., 2018). And the state-of-the-art one is the deep learning-based approach. However, most of these models have some limitations. First, some models can only work on a fixed domain *ontology*, i.e. the slots and values are defined in advance, and can't change dynamically. However, this is not flexible in practice (Xu and Hu, 2018). For example, in the tourist information domain, new restaurants or hotels are often added, which results in the change of the ontology. Second, in many approaches the models for every slot are different. Therefore, the number of parameters is proportional to the number of slots. Third, some models extract features based on text *delexicalisation* (Henderson et al., 2014b), which depends on predefined semantic dictionaries. In large scale domains, it's hard to manually construct the semantic dictionaries for all slots and values (Mrkšić et al., 2017).

To tackle these challenges, here we propose a *universal* dialogue state tracker, StateNet. For each state slot, StateNet generates a fixed-length representation of the dialogue history, and then compares the distances between this representation and the value vectors in the candidate set for making prediction. The set of candidate values

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2780–2786
Brussels, Belgium, October 31 - November 4, 2018.

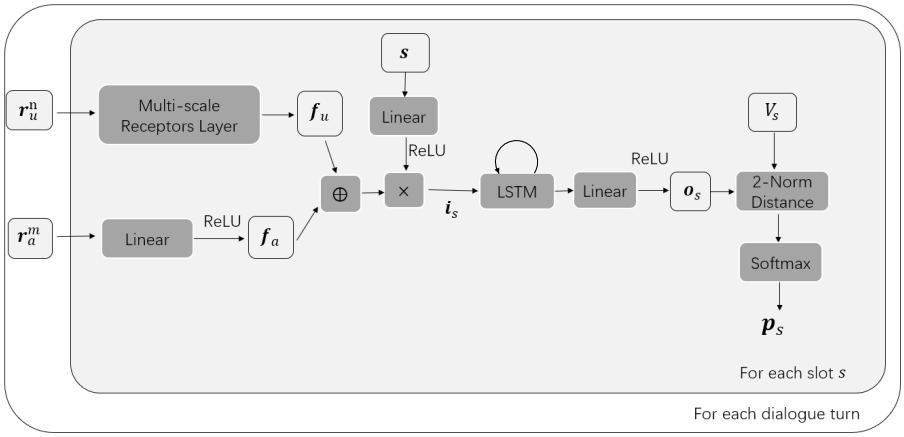

Figure 1: General model architecture of StateNet.

can change dynamically. StateNet only needs the following three parts of the data: (1) the original ASR information (or the transcript) of the user utterance; (2) the information of the machine act; (3) the literal names of the slots and the values. The manually-tagging of the user utterance is not needed as a part of the data. StateNet shares parameters among all slots, through which we can not only transfer knowledge among slots but also reduce the number of parameters.

2 StateNet: A Universal Dialogue State Tracker

For each dialogue turn, StateNet takes the multiple n-gram user utterance representation, $\mathbf{r}_u^n$, the n-gram machine act representation, $\mathbf{r}_a^n$, the value set, $\mathcal{V}_s$, and the word vector of the slot, $\mathbf{s}$, as the input. Then StateNet applies the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) to track the inner dialogue states among the dialogue turns. And for each slot, StateNet outputs a corresponding probability distribution, $\mathbf{p}_s$, over the set of possible values, $\mathcal{V}_s$, at each of the dialogue turn,

$$\mathbf{p}_s = \text{StateNet}(\mathbf{r}_u^n, \mathbf{r}_a^n, \mathbf{s}, \mathcal{V}_s).$$

The general model architecture is shown in Figure 1.

2.1 User Utterance Representation

At the t-th dialogue turn, the user utterance, U_t, may consist of l number of words, u_i, with their corresponding word vectors, $\mathbf{u}_i$, $(1 \leq i \leq l)$. The user utterance may also have its corresponding m-best ASR hypotheses with the normalized confidence scores (Chen et al., 2017), q_j,$(1 \leq j \leq m)$. In this case, we can calculate the weighted word vectors, $\mathbf{u'}_i$,

$$\mathbf{u'}_i = \sum_{j=1}^{m} q_j \mathbf{u}_{i,j},$$

where $\mathbf{u}_{i,j}$ represents the word vector $\mathbf{u}_i$ presented at the j-th ASR hypothesis, and the zero vectors are padded at the end of all the hypotheses that are shorter than the longest one to have a same length of the utterance.

Based on the weighted word vectors generalizing the information from the ASR hypothesis, we can then construct the n-gram weighted word vectors, as proposed by Mrkšić et al. (2017),

$$\mathbf{u'}_i^n = \mathbf{u'}_i \oplus ... \oplus \mathbf{u'}_{i+n-1},$$

where $\oplus$ is the concatenation operator between the word vectors.

An n-gram user utterance representation is then constructed through a sum of the n-gram weighted word vectors,

$$\mathbf{r}_u^n = \sum_{i=1}^{l-n+1} \mathbf{u'}_i^n.$$

2.2 Multi-scale Receptors Layer

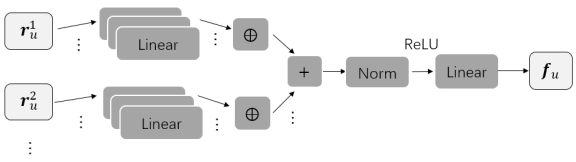

Figure 2: Multi-scale Receptors Layer.

For each gram k of the user utterance representation, $\mathbf{r}_u^k$, $(1 \leq k \leq n)$, the Multi-scale Receptors Layer has c number of linear neural networks

(with the same number of neurons, N_c). Each of them takes the representation as input and is expected to work as the specialized receptor to amplify the signals from some of the word vectors in the utterance representation,

$$\hat{\mathbf{r}}_u^k = \oplus_{j=1}^{c}(\mathbf{W}_k^j \mathbf{r}_u^k + \mathbf{b}_k^j),$$

where $\mathbf{W}_k^j$ means the weight of the j-th linear layer, $\mathbf{b}_k^j$ means the corresponding bias, and $\oplus$ is the concatenation operator between the neurons of these linear layers. Note that each receptor does not necessarily has to be a single linear neural network and can be sophisticated with multiple layers and non-linearity for better detection performance. Here we only use the linear layer to provide a baseline of this kind of structure design.

These c number of linear layers (or receptors) for different grams (or scales) of the representation $\hat{\mathbf{r}}_u^k$ is then summed together to be layer-normalized (Ba et al., 2016). After that, the ReLU activation function is applied, followed by a linear layer with the size N_c that maps all the receptors to a user feature vector, $\mathbf{f}_u$,

$$\mathbf{f}_u = \text{Linear}(\text{ReLU}(\text{LayerNorm}(\sum_{k=1}^{n} \hat{\mathbf{r}}_u^k))).$$

2.3 Machine Act Representation

We represent the machine act in the m order n-gram of bag of words, $\mathbf{r}_a^m$, based on the vocabularies generalized from the machine acts in the training set of a given data set. The machine act feature, $\mathbf{f}_a$, is then simply generated through a linear layer of size N_c with the ReLU activation function,

$$\mathbf{f}_a = \text{ReLU}(\text{Linear}(\mathbf{r}_a^m)).$$

2.4 Slot Information Decoding

Since a slot, e.g. *area* or *food*, is usually indicated as a word or a short word group, then it can be represented as a single word vector (with multiple word vectors summed together), $\mathbf{s}$. A single linear layer with the size $2N_c$ is applied to the word vector $\mathbf{s}$, followed by the ReLU non-linear layer,

$$\mathbf{f}_s = \text{ReLU}(\text{Linear}(\mathbf{s})).$$

The turn-level feature vector, $\mathbf{i}_s$, is then generated through a point-wise multiplication $\otimes$ between the slot feature and the concatenation of the user feature and the machine act feature,

$$\mathbf{i}_s = \mathbf{f}_s \otimes (\mathbf{f}_u \oplus \mathbf{f}_a).$$

In this way, the turn-level feature vector is intended to amplify the large magnitude signals that are from both the user and machine act feature vector and the slot feature vector.

2.5 Fixed-length Value Prediction

Given the turn-level feature vector, $\mathbf{i}_s$, we can now track the dialogue state throughout the dialogue turns by LSTM. For the current turn t, the LSTM takes the $\mathbf{i}_s$ and the previous hidden state, $\mathbf{q}_{t-1}$, as the input. We can then obtain a fixed-length value prediction vector, $\mathbf{o}_s$, whose length is equal to N_w, i.e. the dimension of the word vectors which are fed into the model,

$$\mathbf{o}_s = \text{ReLU}(\text{Linear}(\text{LSTM}(\mathbf{i}_s, \mathbf{q}_{t-1}))),$$

where the linear layer has N_w neurons. In this way, the prediction of the model is independent of the number of the given values, so it is possible for the model to perform parameter sharing among each of the slots. The fixed-length prediction can somehow be interpreted as a *word vector* that is ready for the calculation of the similarity between the prediction and the true value label.

2.6 2-Norm Distance

For a specific semantic slot, since there may be no corresponding value in a given dialogue turn, thus we always add a literally "none" value to the value set for the model to track this state. For the evaluation of the similarity between the prediction and the value, we calculate the 2-Norm distance between the prediction vector and each of the word vectors of the values in the value set. Softmax function is performed with respect to all the negative relative distances to give a distribution of probabilities for the values, $v_i \in \mathcal{V}_s$,

$$p_s(v_i) = \text{Softmax}(-||\mathbf{o}_s - \mathbf{v}_i||),$$

where $\mathbf{v}_i$ is the representation vector of v_i. If the slot value v_i consists of more than one word, $\mathbf{v}_i$ will then be the summation of all corresponding word vectors. When training the model, we minimize the Cross-Entropy (CE) loss between the output probabilities and the given label.

StateNet requires the user utterance, the semantic slots, and slot values to be able to be expressed in words and have their corresponding word vectors. We use the fixed word embedding for every word, and do not fine-tune the word embeddings in the model. Since the word embeddings

are distributed on a fixed-dimension vector space and hold rich semantic information, StateNet may have the ability to track the dialogue state for any new slot or value, as long as the corresponding word embedding can be found. This is the reason why we call the StateNet a *universal* dialogue state tracker.

3 Experiments

Experiments are conducted to assess the performance on joint goal. Two datasets are used by us for training and evaluation. One is the second Dialogue State Tracking Challenge (**DSTC2**) dataset (Henderson et al., 2014a), and the other is the second version of Wizard-of-Oz (**WOZ 2.0**) dataset (Wen et al., 2017). Both of them are the conversations between users and a machine system. The user's goal is to find a suitable restaurant around Cambridge. The ontology of these two datasets is identical, which is composed of three informable slots: *food, pricerange* and *area*. The main difference between them is that in WOZ 2.0, users typed instead of using speech directly. This means the users can use far more sophisticated language than they can in the DSTC2, which is a big challenge for the language understanding ability of the model. Thus, it allows WOZ 2.0 to be more indicative of the model's actual performance since it is immune to ASR errors.

Based on the model structure as described in Section 2, we implement three kinds of dialogue state tracker. The difference among them lies in the utilization of parameter sharing and parameter initialization.

- `StateNet`: It doesn't have shared parameters among different slots. In other words, three models for three slots are trained separately using RMSProp optimizer, learning rate set to 0.0005. And its parameters are not initialized with any pre-trained model.
- `StateNet_PS`: Parameter sharing is conducted among three slots. For each slot in a batch, we infer the model with the slot information and the same dialogue information. The losses are calculated based on the corresponding value set. After each slot is inferred, we back-propagate all the losses and do the optimization. So we just train one model in total using RMSProp optimizer, learning rate set to 0.0005. As a result, the amount of model parameters is one third of that of `StateNet`, which means `StateNet_PS` can significantly save the memory usage during inferring.
- `StateNet_PSI`: Parameter sharing is conducted within this model, same as `StateNet_PS`, but its parameters are initialized with a pre-trained model. For pre-training, we only allow the model to track one single slot and make predictions on its value set. After the training ends, we save the model parameters and use them to initialize the model parameters for the training of the multi-slot tracking. The pre-trained model with the best performance on the validation set is selected for initialization. Here, we choose the *food* slot for pre-training since `StateNet` has the lowest prediction accuracy on the *food* slot. `StateNet_PSI` is trained using Adam optimizer and learning rate is set to 0.001. Since the model has obtained the basic knowledge from the pre-trained model, then a more aggressive learning process is preferred. Adam with a higher learning rate can help a lot compared to RMSProp optimizer.

The hyperparameters are identical for all three models, $N_c = 128, N_w = 300, n = 2, m = 3$. We use $c = 4$ for the number of the receptors for each slot, where the number is determined through the grid search. The word embeddings used by us is the *semantically specialised* Paragram-SL999 vectors (Wieting et al., 2015) with the dimension of 300, which contain richer semantic contents compared to other kinds of word embeddings. Implemented with the MXNet deep learning framework of Version 1.1.0, the model is trained with a batch size of 32 for 150 epochs on a single NVIDIA GTX 1080Ti GPU.

The results in Table 1 show the effectiveness of parameter sharing and initialization. `StateNet_PS` outperforms `StateNet`, and `StateNet_PSI` performs best among all 3 models. It is because the parameter sharing can not only prevent the model diverging from the right learning process but also transfer necessary knowledge among different slots. And the parameter initialization provides the model with the opportunity to gain some basic while essential semantic information at the very beginning since the *food* slot is the most important and difficult one. Besides, `StateNet_PSI` beats all the mod-

DST Models	Joint Acc. DSTC2	Joint Acc. WOZ 2.0
Delexicalisation-Based (DB) Model (Mrkšić et al., 2017)	69.1	70.8
DB Model + Semantic Dictionary (Mrkšić et al., 2017)	72.9	83.7
Scalable Multi-domain DST (Rastogi et al., 2017)	70.3	-
MemN2N (Perez and Liu, 2017)	74.0	-
PtrNet (Xu and Hu, 2018)	72.1	-
Neural Belief Tracker: NBT-DNN (Mrkšić et al., 2017)	72.6	84.4
Neural Belief Tracker: NBT-CNN (Mrkšić et al., 2017)	73.4	84.2
Belief Tracking: Bi-LSTM (Ramadan et al., 2018)	-	85.1
Belief Tracking: CNN (Ramadan et al., 2018)	-	85.5
GLAD (Zhong et al., 2018)	74.5	88.1
StateNet	74.1	87.8
StateNet_PS	74.5	88.2
StateNet_PSI	**75.5**	**88.9**

Table 1: Joint goal accuracy on DSTC2 and WOZ 2.0 test set vs. various approaches as reported in the literature.

els reported in the previous literature, whether the model with delexicalisation (Henderson et al., 2014b,c; Rastogi et al., 2017) or not (Mrkšić et al., 2017; Perez and Liu, 2017; Xu and Hu, 2018; Ramadan et al., 2018; Zhong et al., 2018).

Initialization	Joint Acc. DSTC2	Joint Acc. WOZ 2.0
food	**75.5**	**88.9**
pricerange	73.6	88.2
area	73.5	87.8

Table 2: Joint goal accuracy on DSTC2 and WOZ 2.0 of `StateNet_PSI` using different pre-trained models based on different single slot.

We also test `StateNet_PSI` with different pre-trained models, as shown in Table 2. The fact that the *food* initialization has the best performance verifies our selection of the slot with the worst performance for pre-training. This is because the good performance on joint goal requires a model to make correct predictions on all of the slots. A slot on which the model has the worst accuracy, i.e. the most difficult slot, will dramatically limit the overall model performance on the metric of the joint goal accuracy. Thus, the initialization with a model pre-trained on the most difficult slot can improve the performance of the model on its weakness slot and boost the joint goal accuracy, while the initialization of a strength slot may not help much for the overall accuracy but in turn causes the over-fitting problem of the slot itself.

4 Conclusion

In this paper, we propose a novel dialogue state tracker that has the state-of-the-art accuracy as well as the following three advantages: 1) the model does not need manually-tagged user utterance; 2) the model is scalable for the slots that need tracking, and the number of the model parameters will not increase as the number of the slots increases, because the model can share parameters among different slots; 3) the model is independent of the number of slot values, which means for a given slot, the model can make the prediction on a new value as long as we have the corresponding word vector of this new value. If there are a great number of values for a certain slot, to reduce the computational complexity, we can utilize a fixed-size candidate set (Rastogi et al., 2017), which dynamically changes as the dialogue goes on. Experiment results demonstrate the effectiveness of parameter sharing & initialization.

Our future work is to evaluate the performance of our models in the scenario where there are new slots and more unobserved slot values, and to evaluate the domain-transferring ability of our models.

Acknowledgments

The corresponding author is Kai Yu. This work has been supported by the National Key Research and Development Program of China (Grant No. 2017YFB1002102) and the China NSFC project (No. 61573241). Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Lu Chen, Bowen Tan, Sishan Long, and Kai Yu. 2018. Structured dialogue policy with graph neural networks. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1257–1268.

Zhehuai Chen, Yimeng Zhuang, and Kai Yu. 2017. Confidence measures for ctc-based phone synchronous decoding. In *IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, pages 4850–4854.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014a. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014b. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014c. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Mettew Henderson, Blaise Thomson, and Jason D. Williams. 2014d. The third dialog state tracking challenge. In *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Sungjin Lee. 2013. Structured discriminative model for dialog state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 442–451, Metz, France. Association for Computational Linguistics.

Sungjin Lee and Maxine Eskenazi. 2013. Recipe for building robust spoken dialog state trackers: Dialog state tracking challenge system description. In *Proceedings of the SIGDIAL 2013 Conference*, pages 414–422, Metz, France. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1777–1788.

Julien Perez and Fei Liu. 2017. Dialog state tracking, a machine reading approach using memory network. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 305–314.

Osman Ramadan, Paweł Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 432–437.

Abhinav Rastogi, Dilek Hakkani-Tur, and Larry Heck. 2017. Scalable multi-domain dialogue state tracking. *arXiv preprint arXiv:1712.10224*.

Kai Sun, Lu Chen, Su Zhu, and Kai Yu. 2014a. A generalized rule based tracker for dialogue state tracking. In *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*.

Kai Sun, Lu Chen, Su Zhu, and Kai Yu. 2014b. The SJTU system for dialog state tracking challenge 2. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 318–326, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Kai Sun, Qizhe Xie, and Kai Yu. 2016a. Recurrent polynomial network for dialogue state tracking. *Dialogue & Discourse*, 7(3):65–88.

Kai Sun, Su Zhu, Lu Chen, Siqiu Yao, Xueyang Wu, and Kai Yu. 2016b. Hybrid dialogue state tracking for real world human-to-human dialogues. In *Proc. InterSpeech*, pages 2060–2064, San Francisco, America.

Blaise Thomson and Steve Young. 2010. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, pages 423–432, Metz, France. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M. Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

John Wieting, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Dan Roth. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, Metz, France. Association for Computational Linguistics.

Kaige Xie, Cheng Chang, Liliang Ren, Lu Chen, and Kai Yu. 2018. Cost-sensitive active learning for dialogue state tracking. In *Proceedings of the 19th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 209–213, Melbourne, Australia. Association for Computational Linguistics.

Qizhe Xie, Kai Sun, Su Zhu, Lu Chen, and Kai Yu. 2015. Recurrent polynomial network for dialogue state tracking with mismatched semantic parsers. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 295–304, Prague, Czech Republic. Association for Computational Linguistics.

Puyang Xu and Qi Hu. 2018. An end-to-end approach for handling unknown slot values in dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

Steve Young, Milica Gasic, Blaise Thomson, and Jason D. Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Kai Yu, Lu Chen, Bo Chen, Kai Sun, and Su Zhu. 2014. Cognitive technology in task-oriented dialogue systems: Concepts, advances and future. *Chinese Journal of Computers*, 37(18):1–17.

Kai Yu, Lu Chen, Kai Sun, Qizhe Xie, and Su Zhu. 2016. Evolvable dialogue state tracking for statistical dialogue management. *Frontiers of Computer Science*, 10(2):201–215.

Kai Yu, Kai Sun, Lu Chen, and Su Zhu. 2015. Constrained markov bayesian polynomial for efficient dialogue state tracking. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 23(12):2177–2188.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1458–1467.