END-TO-END JOINT LEARNING OF NATURAL LANGUAGE UNDERSTANDING AND DIALOGUE MANAGER

Xuesong Yang[⋆]*, Yun-Nung Chen*[§]*, Dilek Hakkani-Tür*[†]*, Paul Crook*[∗]*, Xiujun Li*[‡]*, Jianfeng Gao*[‡]*, Li Deng*[‡]

[⋆]University of Illinois at Urbana-Champaign, Urbana, IL, USA
[§]National Taiwan University, Taipei, Taiwan
[†]Google Research, Mountain View, CA, USA
[‡]Microsoft Research, Redmond, WA, USA, [∗]Microsoft Corporation, Redmond, WA, USA

ABSTRACT

Natural language understanding and dialogue policy learning are both essential in conversational systems that predict the next system actions in response to a current user utterance. Conventional approaches aggregate separate models of natural language understanding (NLU) and system action prediction (SAP) as a pipeline that is sensitive to noisy outputs of error-prone NLU. To address the issues, we propose an end-to-end deep recurrent neural network with limited contextual dialogue memory by jointly training NLU and SAP on DSTC4 multi-domain human-human dialogues. Experiments show that our proposed model significantly outperforms the state-of-the-art pipeline models for both NLU and SAP, which indicates that our joint model is capable of mitigating the affects of noisy NLU outputs, and NLU model can be refined by error flows backpropagating from the extra supervised signals of system actions.

Index Terms— language understanding, spoken dialogue systems, end-to-end, dialogue manager, deep learning

1. INTRODUCTION

Recent progress of designing conversational agents for commercial purposes, such as Microsoft's Cortana, Apple's Siri, and Amazon's Echo, has attracted more attention from both academia and industry. Two essential components of these conversational agents are natural language understanding (NLU) and dialog manager (DM). NLU typically detects dialog domains by parsing user utterances followed by user intent classification and filling associated slots according to a domain-specific semantic template [1]; DM keeps monitoring the belief distribution over all possible user states underlying current user behaviors, and predicts responsive system actions [2, 3]. For example, given a user utterance "*any action movies recommended this weekend?*", NLU predicts intent `request_movie` and slots `genre` and `date`; thereafter, DM predicts system action `request_location`.

Traditional approaches for NLU usually model tasks of domain/intent classification and slot filling separately. Sequential labeling methods, such as hidden Markov models (HMMs) and conditional random field (CRF) are widely used in slot tagging tasks [4–6]; maximum entropy and support vector machines with linear kernel (LinearSVM) are applied to user intent prediction [7–9]. These models highly rely on careful feature engineering that is laborious and time-consuming. Deep learning techniques making incredible progress on learning expressive feature representations have achieved better solutions to NLU modeling in ATIS domain [10–14]. The performance was improved significantly by incorporating recurrent neural networks (RNN) and CRF model [15–17]. Convolutional neural networks are also used for domain/intent classification [18, 19].

Slot tags and intents, as semantics representations of user behaviors, may share knowledge with each other such that separate modeling of these two tasks is constrained to take full advantage of all supervised signals. Flexible architectures of neural networks provide a way of jointly training with intent classification and slot filling [20, 21]. Contextual information of previous queries and domain/intent prediction was also incorporated into RNN structures [22, 23].

Information flows from NLU to DM, and noisy outputs of NLU are apt to transfer errors to the following DM, so that it brings in challenges for monitoring the belief distribution and predicting system actions. Most successful approaches cast the dialog manager as a partially observable Markov decision process [2], which uses hand-crafted features to represent the state and action space, and requires a large number of annotated conversations [24] or human interactions [25, 26]. Converting these methods into practice is far from trivial, and exact policy learning is computational intractable. Therefore, they are constrained to narrow domains.

In order to address the above problems, we propose an end-to-end deep RNN with limited contextual dialog memory that can be jointly trained by three supervised signals—user slot tagging, intent prediction and system action prediction (SAP). Our model expresses superb advantages in natural language understanding and dialog manager. Highly expressive feature representations beyond conventional aggregation

of slot tags and intents are expected to be captured in our joint model, so that the affects of noisy output from NLU can be mitigated. Extra supervised signal from system actions is capable of refining NLU model by backpropagating the associated error gradients.

2. END-TO-END JOINT MODEL

The joint model can be considered as a SAP model stacked on top of a history of NLU models (see Fig. 1). NLU model is designed as a multi-tasking framework by sharing bi-directional long short-term memory (biLSTM) layers with slot tagging and intent prediction.

2.1. Sequence to Sequence Model with biLSTM Cells

Given a sequence of input vectors $\mathbf{x}=\{x_t\}_1^T$, a recurrent unit $\mathcal{H}$ computes a sequence of hidden vectors $\mathbf{h}=\{h_t\}_1^T$ and a sequence of output symbols $\hat{\mathbf{y}}=\{\hat{y}_t\}_1^T$ by iterating the following equations,

$$h_t = \mathcal{H}(x_t, h_{t-1}) = \sigma(W_{xh}x_t + U_{hh}h_{t-1})$$
$$\hat{y}_t = \arg\max(\texttt{softmax}(W_{hy}h_t))$$

where $\texttt{softmax}(z_m) = e^{z_m}/\sum_i e^{z_i}$, σ is an activation function, and W_{xh}, U_{hh} and W_{hy} are weight matrices. The goal of sequence to sequence model (Seq2Seq) is to estimate a conditional probability $p(\hat{\mathbf{y}}|\mathbf{x}) = \prod_{t=1}^{T} p(\hat{y}_t|\mathbf{x})$ such that the distance (loss) between predicted distribution $p(\hat{y}_t|\mathbf{x})$ and target distribution $q(y_t|\mathbf{x})$ is minimized, namely,

$$\texttt{loss} = -\sum_{t=1}^{T}\sum_{z=1}^{M} q(y_t = z|\mathbf{x}) \log p(\hat{y}_t = z|\mathbf{x})$$

where M is the number of unique output labels. The loss of this Seq2Seq model can be optimized using backpropagation. LSTM cells are chosen as recurrent units since LSTM can mitigate problems of vanishing or exploding gradients in long-term dependencies via self-regularization [27]. The LSTM recurrent unit $\mathcal{H}$ can be further expanded as,

$$h_t = \mathcal{H}(x_t, h_{t-1}) = o_t \odot \texttt{tanh}(c_t)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$
$$o_t = \texttt{sigm}(W_{xo}x_t + U_{ho}h_{t-1}),\ i_t = \texttt{sigm}(W_{xi}x_t + U_{hi}h_{t-1})$$
$$f_t = \texttt{sigm}(W_{xf}x_t + U_{hf}h_{t-1}),\ g_t = \texttt{tanh}(W_{xg}x_t + U_{hg}h_{t-1})$$

where the sigmoid functions $\texttt{sigm}$ and $\texttt{tanh}$ are applied element-wise, and $\odot$ denotes element-wise product. Since preceding and following lexical contexts are important in analysis of user utterances, bi-directional LSTM cells [28] are used. Therefore sequence $\mathbf{x}$ and its reverse go through LSTM layers separately, followed by the concatination of the corresponding forward output $\overrightarrow{\mathbf{h}}$ and backward output $\overleftarrow{\mathbf{h}}$,

$$\overrightarrow{h}_t = \mathcal{H}\left(x_t, \overrightarrow{h}_{t-1}\right), \quad \overleftarrow{h}_t = \mathcal{H}\left(x_t, \overleftarrow{h}_{t+1}\right)$$
$$\hat{y}_t = \arg\max\left(\texttt{softmax}\left(\overrightarrow{W}_{hy}\overrightarrow{h}_t + \overleftarrow{W}_{hy}\overleftarrow{h}_t\right)\right)$$

where $\overrightarrow{W}_{hy}$ and $\overleftarrow{W}_{hy}$ are bi-directional weight matrices.

2.2. Joint Modeling

The proposed joint model is a RNN classifier that utilizes bi-directional LSTM cells $\mathcal{H}$, which takes as inputs I-1 history of current hidden outputs $\mathbf{h}_j^{(nlu)} = \{h_i^{(nlu)}\}_{j-I+1}^{j}$ from NLU units and performs one-vs-all binary classifications for SAP at the output layer (see Fig. 1), in other word,

$$\overrightarrow{h}_i^{(act)} = \mathcal{H}\left(h_i^{(nlu)}, \overrightarrow{h}_{i-1}^{(act)}\right), \quad \overleftarrow{h}_i^{(act)} = \mathcal{H}\left(h_i^{(nlu)}, \overleftarrow{h}_{i+1}^{(act)}\right)$$
$$p^{(act)} = \texttt{sigm}\left(\overrightarrow{W}_{hy}^{(act)}\overrightarrow{h}_j^{(act)} + \overleftarrow{W}_{hy}^{(act)}\overleftarrow{h}_j^{(act)}\right)$$
$$\hat{y}_k^{(act)} = \begin{cases} 1, & p_k^{(act)} \geq \texttt{threshold} \\ 0, & otherwise \end{cases}$$

where $k \in [1, K]$ denotes the index of system action labels. NLU model at the i-th history is considered as a multi-task joint model with shared biLSTM layers for two tasks, where it takes as inputs a sequence of word vectors $\mathbf{w} = \{w_t\}_1^T$, and performs Seq2Seq for slot tagging and one-vs-all binary classifications for intent prediction (see Fig. 2). The biLSTM architecture mentioned in Section 2.1 can be directly applied to slot tagging task with M unique user slot tags,

$$\overrightarrow{h}_t^{1(i)} = \mathcal{H}\left(w_t, \overrightarrow{h}_{t-1}^{1(i)}\right), \quad \overleftarrow{h}_t^{1(i)} = \mathcal{H}\left(w_t, \overleftarrow{h}_{t+1}^{1(i)}\right)$$
$$\hat{y}_t^{(tag_i)} = \arg\max\left(\texttt{softmax}\left(\overrightarrow{W}_{hy}^{(tag)}\overrightarrow{h}_t^{1(i)} + \overleftarrow{W}_{hy}^{(tag)}\overleftarrow{h}_t^{1(i)}\right)\right)$$

where $\overrightarrow{h}_t^{1(i)}$ and $\overleftarrow{h}_t^{1(i)}$ denotes hidden outputs of the shared forward and backward layers, respectively. As for intent pre-

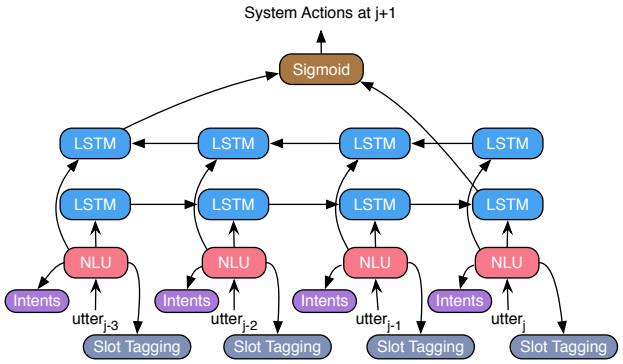

Fig. 1: Proposed end-to-end joint model

diction task in NLU, we add one more recurrent LSTM layer on top of biLSTM layers, and only consider the last hidden vector $h_T^{2(int_i)}$ as the output of this second recurrent layer. Real human-human dialogs encode various number of intents in a single user utterance, and therefore, we design a set of one-vs-all binary classifiers at the output layer where each neuron is activated using a sigmoid function. The positive label of each classifier is predicted if its probability is no less

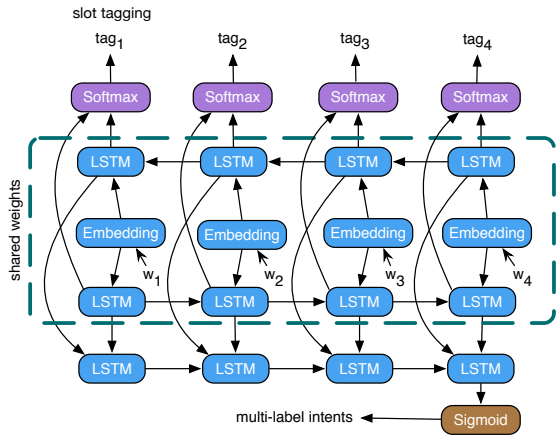

Fig. 2: biLSTMs-based NLU model

than the threshold,

$$h_t^{2(int_i)} = \mathcal{H}\left(h_{t-1}^{2(int_i)}, \overrightarrow{h}_t^{1(i)}, \overleftarrow{h}_t^{1(i)}\right)$$

$$p^{(int_i)} = \texttt{sigm}\left(W_{hy}^{2(int)} h_T^{2(int_i)}\right)$$

$$\hat{y}_n^{(int_i)} = \begin{cases} 1, & p_n^{(int_i)} \geq \texttt{threshold} \\ 0, & otherwise \end{cases}$$

where $n \in [1, N]$ is the index of intent labels. We choose the same two-layer recurrent architecture as the intent model to calculate the hidden vector $h_i^{(nlu)}$ out from the i-th NLU component with the size of M+N, where M and N are the number of unique slot tags and unique intents, respectively.

$$h_i^{(nlu)} = h_T^{2(nlu_i)}, \quad h_t^{2(nlu_i)} = \mathcal{H}\left(h_{t-1}^{2(nlu_i)}, \overrightarrow{h}_t^{1(i)}, \overleftarrow{h}_t^{1(i)}\right)$$

End-to-end joint training estimates the conditional probability given a history of word vectors $\mathbf{w}_h = \{\mathbf{w}^{(i)}\}_1^I$ such that $\texttt{loss} = l^{(act)} + l^{(tag)} + l^{(int)}$ is minimized, where

$$l^{(act)} = -\sum_{k=1}^{K}\sum_{z=0}^{1} q\left(y_k^{(act)} = z \mid \mathbf{w}_h\right) \log p\left(\hat{y}_k^{(act)} = z \mid \mathbf{w}_h\right)$$

$$l^{(tag)} = -\sum_{i=1}^{I}\sum_{t=1}^{T}\sum_{z=1}^{M} q\left(y_t^{(tag_i)} = z|\mathbf{w}^{(i)}\right) \log p\left(\hat{y}_t^{(tag_i)} = z|\mathbf{w}^{(i)}\right)$$

$$l^{(int)} = -\sum_{i=1}^{I}\sum_{n=1}^{N}\sum_{z=0}^{1} q\left(y_n^{(int_i)} = z \mid \mathbf{w}^{(i)}\right) \log p\left(\hat{y}_n^{(int_i)} = z \mid \mathbf{w}^{(i)}\right)$$

3. EXPERIMENTS

3.1. Training Configurations

We choose a mini-batch stochastic gradient descent method Adam [29] with the batch size of 32 examples. The size of each hidden recurrent layer is 256, and the size of hidden output vector of NLU units is M+N, where M and N are the size of unique slot tags and intents, respectively. We assume the joint model can only get access to previous history with I=5. The dimension of word embeddings is 512. Dropout rate is 0.5. We apply 300 training epochs without using any early stop strategy. Best models for three tasks are selected separately upon decision thresholds well tuned on dev set under different metrics. Token-level micro-average F1 score is used for slot filling; frame-level accuracy (it counts only when the whole frame parse is correct) is used for user intent prediction and system action prediction. The code is released[1].

Table 1: *Statistics of data used in experiments. '#' represents the number of unique items.*

	#utters	#words	#tags	#intents	#actions
train	5,648	2,252	87	68	66
dev	1,939	1,367	79	54	53
test	3,178	1,752	75	58	58

Table 2: *Performance (%) of end-to-end models for SAP. F1, P, R are micro-averaged token-level scores; FrmAcc is frame-level accuracy. Oracle models are provided as references.*

Models	F1	P	R	FrmAcc
Baseline (CRF+SVMs)	31.15	29.92	32.48	7.71
Pipeline (biLSTMs)	19.89	14.87	30.01	11.96
JointModel	19.04	18.53	19.57	**22.84**
Oracle-SAP (SVMs)	30.61	30.20	31.04	7.65
Oracle-SAP (biLSTM)	23.09	22.24	24.01	19.67

3.2. Corpus

DSTC4 corpus[2] is selected, which collected human-human dialogs of tourist information in Singapore from Skype calls that spanned five domains—accommodation, attraction, food, shopping, and transportation. Each tourist and guide tend to be expressed in a series of multiple turns. The guide is defined as the system in this paper. We transform raw data into examples that fit our experiments. Each example includes an user utterance and its associated slot tags in IOB format [30], user intents, and responsive system actions. Labels of system actions are defined as the concatenation of categories and attributes of speech acts, e.g. `QST_WHEN`. `NULL` is added as a waiting response from guides when they are expressed in multiple turns. The consecutive guide actions in response to a single tourist utterance is merged as multiple labels. The whole corpus is split into train/dev/test (see Table 1). Unseen tokens such as words, user intents, slot tags, and system actions in the dev/test set are categorized as `UNK`.

3.3. Evaluation Results

We compare our proposed joint model with following models in three tasks: slot filling, intent prediction and SAP.

- Baseline (CRF+SVMs): NLU and SAP are trained separately, followed by being pipelined for testing.

[1] https://github.com/XuesongYang/end2end_dialog.git
[2] http://www.colips.org/workshop/dstc4/data.html

Table 3: *Performance (%) of NLU models, where F1, Precision and Recall are at token-level and FrmAcc is at frame-level.*

Models	User Slot Tagging (UST)				User Intent Prediction (UIP)				NLU (UST+UIP)
	F1	Precision	Recall	FrmAcc	F1	Precision	Recall	FrmAcc	FrmAcc
NLU-Baseline	40.50	61.41	30.21	**77.31**	49.75	52.56	47.24	37.19	33.13
NLU-Pipeline	46.15	54.63	39.96	76.84	47.48	52.19	43.55	39.96	36.38
NLU-JointModel	45.04	53.35	38.97	76.49	49.67	52.22	47.35	**42.20**	**37.38**

CRF is used to train slot filling model with lexical feature of words; one-vs-all SVMs with linear kernel (LinearSVMs) is used to train intent model with bag-of-words features of user utterances; SAP utilizes LinearSVMs with features of one-hot vectors of aggregated user slot tags and intents. Decision thresholds for intent model and SAP are 0.225 and 0.162.

- Pipeline (biLSTMs): NLU in Fig. 2 and SAP in Fig. 3 are separately trained, followed by being pipelined for testing. Best decision thresholds for intent model and SAP model are 0.391 and 0.064.
- Oracle-SAP (SVMs): The inputs of SAP are clean slot tags and intents annotated by human experts; LinearSVMs is used for training and testing SAP. Best decision threshold is 0.162.
- Oracle-SAP (biLSTM): SAP takes as inputs the same to Oracle-SAP but uses biLSTM for training and testing (see Fig. 3). Best decision threshold is 0.064.

Evaluation results of end-to-end models are illustrated in Table 2. Our proposed joint model outperforms all other end-to-end models in frame-level accuracy by a large margin. The joint model and biLSTMs pipeline achieved absolute increase over baseline with 15.03% and 4.25%, respectively. Both models beat the SVMs oracle scores. The biLSTMs pipeline model get worse than biLSTM oracle as expected since it transfer the errors from NLU to the SAP model. Nevertheless, the joint model obtains 10.88% increase than pipeline model and 3.17% than biLSTM oracle. These promising improvements indicate that joint training can mitigate the downside of pipeline model in that the hidden outputs from a history of NLU units capture highly more expressive feature representations than the conventional aggregation of user intents and slot tags. In comparison of these two oracle models, the large improvement (12.02%) for biLSTM model indicates that the contextual user turns make significant contribution to system action prediction. In real human interaction scenarios, frame-level metrics are far more important than token-level ones especially for these multi-label classification tasks since predicting precise number of labels is more challenging.

Evaluation results of NLU models that are frozen as independent models are illustrated in Table 3. Baseline using CRF and SVMs still maintains a strong frame-level accuracy with 33.13%, however, biLSTM models taken from pipeline and joint model achieve better increase 3.25% and 4.25%, respectively. This observation indicates that joint training with

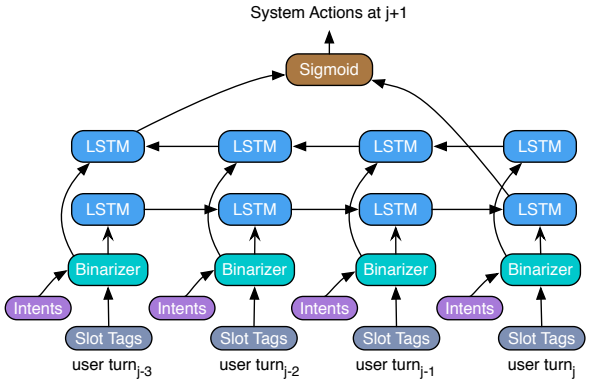

Fig. 3: biLSTM-based SAP model

two tasks of slot filling and intent prediction captures implicit knowledge underlying the shared user utterances, while another supervised signal from system actions is capable of refining the biLSTM based model by backpropagating the associated error gradients. Best accuracy at frame-level for slot filling task is obtained by traditional CRF baseline with only lexical features of words, and our biLSTM models fall behind with absolute decrease 0.47% and 0.82%. Best frame accuracy for intent prediction task is achieved by our proposed model with 5.21% improvement.

4. CONCLUSION

We proposed an end-to-end deep recurrent neural network with limited contextual dialog memory that can be jointly trained by three supervised signals of user slot filling, intent prediction and system action prediction. Experiments on multi-domain human-human dialogs demonstrated that our proposed model expressed superb advantages in natural language understanding and dialog manager. It achieved better frame-level accuracy significantly than the state of the art that pipelines separate models of NLU and SAP together. The promising performance illustrated that contextual dialog memory made significant contribution to dialog manager, and highly expressive feature representations beyond conventional aggregation of slot tags and intents could be captured in our joint model such that the affects of noisy output from NLU were mitigated. Extra supervised signal from system actions is capable of refining NLU model by backpropagating.

References

[1] Gokhan Tür and Renato De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*, John Wiley & Sons, 2011.

[2] Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams, "POMDP-based statistical spoken dialog systems: A review," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.

[3] 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," *arXiv preprint arXiv:1609.00777*, 2016.

[4] Ye-Yi Wang, Li Deng, and Alex Acero, "Spoken language understanding," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 16–31, 2005.

[5] Christian Raymond and Giuseppe Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *INTERSPEECH*, 2007.

[6] John Lafferty, Andrew McCallum, and Fernando Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001, vol. 1, pp. 282–289.

[7] Patrick Haffner, Gokhan Tur, and Jerry H Wright, "Optimizing SVMs for complex call classification," in *ICASSP*. IEEE, 2003, vol. 1, pp. I–632.

[8] Xuesong Yang, Anastassia Loukina, and Keelan Evanini, "Machine learning approaches to improving pronunciation error detection on an imbalanced corpus," in *SLT*, 2014, pp. 300–305.

[9] Ciprian Chelba, Milind Mahajan, and Alex Acero, "Speech utterance classification," in *ICASSP*, 2003, vol. 1, pp. I–280.

[10] Ruhi Sarikaya, Geoffrey E Hinton, and Bhuvana Ramabhadran, "Deep belief nets for natural language call-routing," in *ICASSP*, 2011, pp. 5680–5683.

[11] Ruhi Sarikaya, Geoffrey E Hinton, and Anoop Deoras, "Application of deep belief networks for natural language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 778–784, 2014.

[12] Zhangyang Wang, Yingzhen Yang, Shiyu Chang, Qing Ling, and Thomas S. Huang, "Learning a deep ℓ_∞ encoder for hashing," in *IJCAI*, 2016.

[13] Yun-Nung Chen, Dilek Hakkani-Tur, Gokhan Tur, Asli Celikyilmaz, Jianfeng Gao, and Li Deng, "Knowledge as a teacher: Knowledge-guided structural attention networks," *arXiv preprint arXiv:1609.03286*, 2016.

[14] Yun-Nung Chen, Dilek Hakkani-Tür, Gokan Tur, Asli Celikyilmaz, Jianfeng Gao, and Li Deng, "Syntax or semantics? knowledge-guided joint semantic frame parsing," in *SLT*, 2016.

[15] Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi, "Spoken language understanding using long short-term memory neural networks," in *SLT*, 2014, pp. 189–194.

[16] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig, "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 530–539, 2015.

[17] Kaisheng Yao, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao, "Recurrent conditional random field for language understanding," in *ICASSP*, 2014, pp. 4077–4081.

[18] Puyang Xu and Ruhi Sarikaya, "Contextual domain classification in spoken language understanding systems using recurrent neural network," in *ICASSP*, 2014, pp. 136–140.

[19] Gokhan Tur, Li Deng, Dilek Hakkani-Tür, and Xiaodong He, "Towards deeper understanding: Deep convex networks for semantic utterance classification," in *ICASSP*, 2012, pp. 5045–5048.

[20] Aditya Bhargava, Asli Celikyilmaz, Dilek Hakkani-Tür, and Ruhi Sarikaya, "Easy contextual intent prediction and slot detection," in *ICASSP*, 2013, pp. 8337–8341.

[21] Puyang Xu and Ruhi Sarikaya, "Convolutional neural network based triangular CRF for joint intent detection and slot filling," in *ASRU*, 2013, pp. 78–83.

[22] Yangyang Shi, Kaisheng Yao, Hu Chen, Yi-Cheng Pan, Mei-Yuh Hwang, and Baolin Peng, "Contextual spoken language understanding using recurrent neural networks," in *ICASSP*, 2015, pp. 5271–5275.

[23] Yun-Nung Chen, Dilek Hakkani-Tür, Gokhan Tur, Jianfeng Gao, and Li Deng, "End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding," in *INTERSPEECH*, 2016.

[24] Matthew Henderson, Blaise Thomson, and Steve Young, "Word-based dialog state tracking with recurrent neural networks," in *SIGDIAL*, 2014, pp. 292–299.

[25] Milica Gašić, Catherine Breslin, Matthew Henderson, Dongho Kim, Martin Szummer, Blaise Thomson, Pirros Tsiakoulis, and Steve Young, "On-line policy optimisation of Bayesian spoken dialogue systems via human interaction," in *ICASSP*, 2013, pp. 8367–8371.

[26] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young, "A network-based end-to-end trainable task-oriented dialogue system," *arXiv preprint arXiv:1604.04562*, 2016.

[27] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] Mike Schuster and Kuldip K Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[29] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] Dilek Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang, "Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM," in *INTERSPEECH*, 2016.