SPEAKER-SENSITIVE DUAL MEMORY NETWORKS FOR MULTI-TURN SLOT TAGGING

Young-Bum Kim[*], *Sungjin Lee*[**],*Ruhi Sarikaya*[*]

Amazon Alexa Brain, Seattle, WA[*],
Microsoft Research, Redmond, WA[**]

ABSTRACT

In multi-turn dialogs, natural language understanding models can introduce obvious errors by being blind to contextual information. To incorporate dialog history, we present a neural architecture with *Speaker-Sensitive Dual Memory Networks* which encode utterances differently depending on the speaker. This addresses the different extents of information available to the system - the system knows only the surface form of user utterances while it has the exact semantics of system output. We performed experiments on real user data from Microsoft Cortana, a commercial personal assistant. The result showed a significant performance improvement over the state-of-the-art slot tagging models using contextual information.

1. INTRODUCTION

Major technology companies have been investing in natural language understanding systems as effective tools for human-computer communication [1, 2, 3, 4, 5, 6, 7, 8]. Of particular interest are task-oriented personal assistants, e.g., Amazon's Alexa, Apple Siri and Microsoft Cortana. In task-oriented dialogs, the main role of the system is to fill out the associated slots to the semantic frame of the requested task. As the demand for fulfilling more complex tasks with voice-interface agents keeps increasing, it becomes crucial to address some issues faced when handling multi-turn interaction. For example, in Table 1, the user's utterance at turn 3 could mean either *number_people* or *time* without any context.

Traditionally, the natural language understanding (NLU) module [9] delegates the task of contextual interpretation to downstream modules such as dialog state tracker [10]. However, given that most dialog state trackers just exploit the slot-values recognized by the NLU module as is without further transformation based on context, the NLU errors due to the context-insensitivity are very likely to propagate through downstream modules. Thus it is critical to reduce the context-related NLU errors as early in the system pipeline as possible.

With this aim, a host of previous studies has attempted to incorporate dialog history for contextual interpretation [11, 12, 13]. A large body of work, however, exploited only previous turn information such as the NLU results at the previous turn. Not only are they limited in the range of history information but also they internally suffer the error propagation problems by directly using previous predictions. With the rise of end-to-end deep neural network models, Chen et al. [14] addressed such shortcomings with end-to-end memory networks [15] where the embeddings of all past user utterances are stored in a memory. In their work, however, no system's information was used which will prove powerful in our experimental results.

In this paper we propose an end-to-end neural architecture for slot tagging that uses *Speaker-Sensitive Dual Memory Networks* (SSDMNs) to leverage both user and system information in a tailored manner. Specifically, since the system only gets to see the surface form of user's utterance, we encode each word of all past user's utterances using the Long Short-Term Memory (LSTM) model [16]. We store the encoded vectors of each word in a memory and use an attention mechanism [17] to selectively use only relevant words to the current user input. Whereas the system already knows the exact semantics of its past utterances, thus we directly encode the targeted slots of system's question to avoid parsing noisy natural language expressions. Due to this different nature, we introduce a separate memory storing all system's targeted slot embeddings. The attention mechanism again allows to selectively use relevant system information. Hori et al. [18] adopted a similar concept, role-dependent encoding, but their work is not aimed for slot tagging, nor does it exploit the system output semantics. Furthermore, we adopt an efficient pre-training method [19, 20, 21, 22] for system's targeted slot encoding that further improves the

Turn	Speaker	Utterance	Targeted Slots
1	System	Where do you want to reserve?	*place_name*
	User	Pizza Hut	-
2	System	For how many people?	*number_people*
	User	Two	-
3	System	Okay. What time should I make the reservation for?	{*time, date*}
	User	Actually it's three	-

Table 1. An example dialog for restaurant reservation. The targeted slots are the slots that the system asks about at each turn.

system's performance. To verify the efficacy of the proposed model, we performed evaluation on a large amount of real data gathered from five different domains of commercial personal assistants. In comparison to previous approaches our method showed a large improvement.

The rest of the paper is structured as follows. In Section 2 we describe SSDMNs in depth. In Section 3 we discuss our experiments. We finish with conclusions in Section 4.

2. SPEAKER-SENSITIVE DUAL MEMORY NETWORKS

At turn t, the multi-turn slot tagging model takes a user input $\boldsymbol{u}^t$ and outputs semantic tags $\boldsymbol{m}^t$ by considering the contextual information, i.e., a sequence of user's past utterances $\{\boldsymbol{u}^1, \dots, \boldsymbol{u}^{t-1}\}$ and the corresponding system outputs $\{\boldsymbol{m}^1, \dots, \boldsymbol{m}^{t-1}\}$.

2.1. Overall architecture

The overall SSDMNs architecture is illustrated in Figure 1. The model first encodes the last user input and system output to produce vector embeddings. Then it updates the dual memories with the generated vectors. After that, as it encodes the current user utterance, at each word position, it computes the attention weights to find relevant information. Finally it makes prediction for each word by jointly considering the encoded vector of each word and the memory content averaged with respect to the attention weights. The main steps are described below.

2.2. User utterance encoding

In order to obtain a more contextful vector representation of the j-th word u^i_j of the user utterance at turn i, we adopt a stateful LSTM instead of a vanilla LSTM encoder - the LSTM output for the last word in an utterance is fed as the initial state into the LSTM network for the next utterance while all LSTM weights are shared. The stateful LSTM runs over both past user utterances and the current input:

$$\tilde{\boldsymbol{u}}^i = \text{StatefulLSTM}(\boldsymbol{u}^i), i < t$$

$$\boldsymbol{h}^t = \text{StatefulLSTM}(\boldsymbol{u}^t)$$

2.3. System output encoding

Since the system has the exact semantics of system utterances, we directly encode the targeted slots that usually have strong influence on the following user's response. Concretely, for each system output, we are given an k-hot vector $\mathbf{m}^i \in \mathbb{R}^l$, where l is the number of slot tags, which indicates which slot is targeted by the system's output. Then we use a trainable projection matrix $\mathbf{P} \in \mathbb{R}^{d \times l}$ to map the k-hot vector to a d-dimensional dense vector $\tilde{\mathbf{m}}^i$:

$$\tilde{\mathbf{m}}^i = \mathbf{P}\mathbf{m}^i$$

Note that since $\mathbf{m}^i$ is a k-hot vector, $\tilde{\mathbf{m}}^i$ is the sum of the targeted slot embeddings. It has been observed that pre-trained embeddings can serve as good initialization values that help avoid converging to bad local optima and speed up model training. Inspired by [19], we adopt *Canonical Correlation Analysis* (CCA) to efficiently pretrain the projection matrix. The CCA algorithm finds a lower dimensional projection that maximizes the correlation between a set of random vector pairs:

- $\{\mathbf{a}^i\}$ is a set of zero vectors in which the entry corresponding to the slot tag of the i-th sentence is set to 1.
- $\{\mathbf{b}^i\}$ is a set of zero vectors in which the entries corresponding to words spanned by the tag are set to 1.

2.4. Memory update

Although an LSTM network can capture long-term dependencies, it has been argued that LSTM networks'

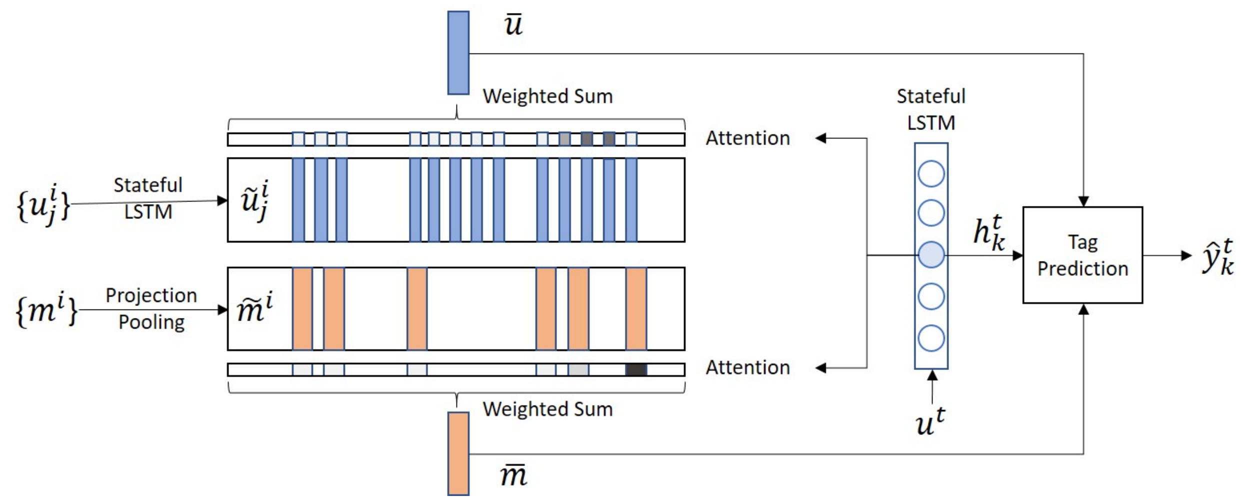

Fig. 1. The overall architecture of *Speaker-Sensitive Dual Memory Networks*.

fixed-size memory may not be enough to contain a large amount of information encoded in variable-sized inputs [17]. Thus before performing the tag prediction we update the dual memories with the encoded vectors of each word of the user utterances ($\tilde{\mathbf{u}}_k^i$) and system output vectors ($\tilde{\mathbf{m}}^i$) encoded in the manner described above, respectively.

2.5. Attention mechanism

Since not every piece of history information is helpful in making prediction for a current word, we selectively use only relevant information. To this end, we first compute the attention weights for both memories:

$$\alpha_j^i = \mathrm{softmax}(\mathbf{W}_\alpha(\tilde{\mathbf{u}}_j^i \circ \mathbf{h}_k^t) + b_\alpha)$$

$$\beta^i = \mathrm{softmax}(\mathbf{W}_\beta(\tilde{\mathbf{m}}^i \circ \mathbf{h}_k^t) + b_\beta)$$

where $\mathbf{h}_k^t$ is the LSTM output for $\mathbf{u}_k^t$, $\mathbf{W}_\alpha$ and $\mathbf{W}_\beta$ are weight matrices, b_α and b_β are bias vectors, $\circ$ denotes element-wise vector multiplication, and $\mathrm{softmax}(x_j^i) = e^{x_j^i}/\sum_{k,l} e^{x_l^k}$. Then we compute the weighted sum of memory entries:

$$\bar{\mathbf{u}} = \sum_{i,j} \alpha_j^i \tilde{\mathbf{u}}_j^i$$

$$\bar{\mathbf{m}} = \sum_i \beta^i \tilde{\mathbf{m}}^i$$

Domain	Train	Dev	Test
Orderfood	5,993	1,333	1,714
Reservation	5,130	1,215	1,339
Taxi	6,120	1,216	1,414
Events	4,347	822	850
Movieticket	3,781	712	738
Average	5,074	1,059	1,211

Table 2. Data description

2.6. Tag prediction

Finally we use $\bar{\mathbf{u}}$ and $\bar{\mathbf{m}}$ along with $\mathbf{h}_k^t$ to make the tag prediction:

$$\hat{\mathbf{y}}_{k,l}^t = \frac{\exp \mathbf{z}_{k,l}^t}{\sum_{l'} \exp \mathbf{z}_{k,l'}^t}$$

where $\mathbf{z}_k^t = \mathbf{W}_h \mathbf{h}_k^t + \mathbf{W}_u \bar{\mathbf{u}} + \mathbf{W}_m \bar{\mathbf{m}}$ and $\mathbf{W}_*$ are all trainable weight matrices.

3. EXPERIMENTS

3.1. Models

To evaluate the SSDMNs model, we conducted comparative experiments with the following systems:

- **LSTM** uses the stateful LSTM only with no contextual information used.
- **+LS** additionally uses the last system output.
- **+PreLS** uses the pre-trained system tag projection matrix for encoding the last system output to show the efficacy of the CCA pre-training scheme.

Domain	LSTM	+LS	+PreLS	MNs	MNs+S	SSDMNs
Orderfood	87.38	93.70	94.10	92.07	94.36	**96.45**
Reservation	86.29	91.96	92.81	90.80	91.64	**94.37**
Taxi	89.46	93.77	94.58	92.57	93.11	**96.67**
Events	85.17	91.39	90.80	87.61	89.64	**92.21**
Movieticket	88.06	93.76	94.53	90.96	91.89	**96.14**
Average	87.27	92.92	93.36	90.80	92.13	**95.17**

Table 3. Comparative performance in F1 score on five domains.

- **MNs** similarly to the Memory Networks model in Chen et al. [14], the system does not use system's output information.
- **MNs+S** MNs with system's output in surface form.

3.2. Data

The data is collected from five domains of Microsoft Cortana. Table 2 shows the number of utterances of the training, development and test datasets of each domain. All datasets were labeled by expert editors.

3.3. Training setting

In our experiments, all the models were implemented using Dynet [23] and were trained using Stochastic Gradient Descent (SGD) with Adam [24]—an adaptive learning rate algorithm. We used the initial learning rate of 4×10^{-4} and left all the other hyper parameters as suggested in [24]. Each SGD update was computed with Intel MKL (Math Kernel Library) [1] without minibatching. We used the dropout regularization [25] with the keep probability of 0.4. Both dimensions of the input and output of the stateful LSTM model were 100. To initialize word embedding, we used the pre-trained GloVe model [26]. For system slot tag embedding, we used CCA to induce 100 dimensional vector representation from 20 personal assistant domains consisting of 3M utterances and 130 distinct tags.

3.4. Results

To compute slot F1-score, we used the standard CoNLL evaluation script[2]. In Table 3, the stateful LSTM-only model (LSTM) achieved 87.27% in averaged F1 score. The improved performance of the +LS model, 92.92%, confirms the same findings of previous studies on the effectiveness of previous turn information for contextual interpretation. The further performance increase of the +PreLS model, achieving 93.36%, demonstrates the possible gain with the proposed pre-training scheme. Interestingly the MNs model couldn't show a better performance than the +LS model, even after incorporating system's output in surface form, justifying our novel contribution on the dual encoding of system output. This claim is also supported by the best performance of the SSDMNs model reaching 95.17%.

4. CONCLUSION

We presented a novel *Speaker-Sensitive Dual Memory Networks* model for the multi-turn slot tagging task which extends the state-of-the-art Memory Networks-based approach with tailored memories to exploit the exact semantic knowledge on system output. We also described an efficient pre-training scheme to obtain better slot tag embeddings. The large performance improvement on real data demonstrates the efficacy of SSDMNs. Future work includes a detailed analysis on various reasoning behaviors of the model depending on different contexts. Also we plan to incorporate more elements of dialog state beyond system output. Another interesting direction is to perform multiple memory references to handle more complex discourse phenomena such as ellipsis, anaphora resolution.

5. REFERENCES

[1] Young-Bum Kim, Minwoo Jeong, Karl Stratos, and Ruhi Sarikaya, "Weakly supervised slot tagging with partially labeled sequences from web search click logs," in *Proceedings of the NAACL*. Association for Computational Linguistics, 2015.

[2] Young-Bum Kim, Karl Stratos, Xiaohu Liu, and Ruhi Sarikaya, "Compact lexicon selection with spectral methods," in *Proc. of Annual Meeting of*

[1] https://software.intel.com/en-us/articles/intelr-mkl-and-c-template-libraries

[2] http://www.cnts.ua.ac.be/conll2000/chunking/output.html

the Association for Computational Linguistics: Human Language Technologies, 2015.

[3] Ruhi Sarikaya, Paul Crook, Alex Marin, Minwoo Jeong, Jean-Philippe Robichaud, Asli Celikyilmaz, Young-Bum Kim, Alexandre Rochette, Omar Zia Khan, Xiuahu Liu, et al., "An overview of end-to-end language understanding and dialog management for personal digital assistants," in *IEEE Workshop on Spoken Language Technology*, 2016.

[4] Young-Bum Kim, Alexandre Rochette, and Ruhi Sarikaya, "Natural language model re-usability for scaling to different domains," in *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2016.

[5] Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya, "Frustratingly easy neural domain adaptation," *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, 2016.

[6] Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya, "Domainless adaptation by constrained decoding on a schema lattice," *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, 2016.

[7] Young-Bum Kim, Karl Stratos, and Dongchan Kim, "Domain attention with an ensemble of experts," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 2017, pp. 643–653, Association for Computational Linguistics.

[8] Young-Bum Kim, Karl Stratos, and Dongchan Kim, "Adversarial adaptation of synthetic or stale data," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. July 2017, pp. 1297–1307, Association for Computational Linguistics.

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

[10] Jason Williams, Antoine Raux, and Matthew Henderson, "The dialog state tracking challenge series: A review," *Dialogue & Discourse*, vol. 7, no. 3, pp. 4–33, 2016.

[11] Aditya Bhargava, Asli Celikyilmaz, Dilek Hakkani-Tür, and Ruhi Sarikaya, "Easy contextual intent prediction and slot detection," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8337–8341.

[12] Puyang Xu and Ruhi Sarikaya, "Contextual domain classification in spoken language understanding systems using recurrent neural network," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 136–140.

[13] Yangyang Shi, Kaisheng Yao, Hu Chen, Yi-Cheng Pan, Mei-Yuh Hwang, and Baolin Peng, "Contextual spoken language understanding using recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5271–5275.

[14] 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 *Proceedings of Interspeech*, 2016.

[15] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al., "End-to-end memory networks," in *Advances in neural information processing systems*, 2015, pp. 2440–2448.

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

[17] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[18] Chiori Hori, Takaaki Hori, Shinji Watanabe, and John R Hershey, "Context-sensitive and role-dependent spoken language understanding using bidirectional and attention lstms," *Interspeech 2016*, pp. 3236–3240, 2016.

[19] Young-Bum Kim, Karl Stratos, Ruhi Sarikaya, and Minwoo Jeong, "New transfer learning techniques for disparate label sets," *ACL. Association for Computational Linguistics*, 2015.

[20] Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya, "Pre-training of hidden-unit crfs," in *Proc. of Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2015.

[21] Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya, “Scalable semi-supervised query classification using matrix sketching,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. August 2016, pp. 8–13, Association for Computational Linguistics.

[22] Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya, “A framework for pre-training hidden-unit conditional random fields and its extension to long short term memory networks,” *Computer Speech & Language*, vol. 46, pp. 311–326, 2017.

[23] Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al., “Dynet: The dynamic neural network toolkit,” *arXiv preprint arXiv:1701.03980*, 2017.

[24] Diederik Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *The International Conference on Learning Representations (ICLR).*, 2015.

[25] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting.,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] Jeffrey Pennington, Richard Socher, and Christopher D Manning, “Glove: Global vectors for word representation,” *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, vol. 12, pp. 1532–1543, 2014.