End-to-end Gated Self-attentive Memory Network for Dialog Response Selection[*]

Shuo Sun[†]
CLSP
Johns Hopkins University
ssun32@jhu.edu

Yik-Cheung Tam[†]
WeChat AI - Pattern Recognition Center
Tencent Inc.
wilsontam@tencent.com

Jie Cao[†]
School of Computing
University of Utah
jcao@cs.utah.edu

Canxiang Yan
WeChat AI - Pattern Recognition Center
Tencent Inc.
chriscxyan@tencent.com

Zuohui Fu
Department of Computer Science
Rutgers University
zuohui.fu@rutgers.edu

Cheng Niu and **Jie Zhou**
WeChat AI - Pattern Recognition Center
Tencent Inc.
niucheng@tencent.com
withtomzhou@tencent.com

Abstract

This paper presents approaches for the noetic end-to-end response selection challenge in DSTC7. Given a pool of response candidates in a dialog history with external domain knowledge, we propose a Gated Self-attentive Memory Network to encode dialog history and external domain knowledge in an end-to-end trainable manner. Our novelty is that each utterance in the memory is enhanced with self-attention building the connection between dialog history and external domain knowledge in a gated multi-hop manner. We ensemble various gated self-attentive memory network with hierarchical GRU baseline models for final submission. Official evaluation results show that our approach ranks at the second place for both student advising and Ubuntu subtasks integrated with external domain knowledge.

Introduction

Contextual modeling is one of the crucial issue in spoken dialog system. In noetic end-to-end response selection challenge in DSTC7 (Yoshino et al. 2018), dialog context includes utterances from multi-turn dialog history and external domain knowledge. For example in Ubuntu corpus, external domain knowledge may be represented as manual pages for Linux commands. In student advising corpus, external domain knowledge may be represented as course descriptions, student profiles etc. Intuitively, dialog context provides useful information to judge whether the next response candidate is relevant to the current dialog context. Our challenge is to encode them effectively in an end-to-end trainable manner.

In this paper, we propose Gated Self-attentive Memory Network (GSMN) for contextual modeling in dialog. We first encode each utterance in a dialog history and response candidates using bi-directional Gated Recurrent Unit (bi-GRU) (Chung et al. 2014) or bi-directional LSTM (Hochreiter and Schmidhuber 1997). We take the concatenation of the first and the last hidden vectors to represent an utterance. Such collection of utterance representation is interpreted as memory (Sukhbaatar, Weston, and Fergus 2015; Chen et al. 2016; Liu and Perez 2017). External domain knowledge in the DSTC7 challenge contains a set of key-value pairs for an entity subject. In student advising corpus as an example, a course ID “EECS183” has a key “Course Title” with a value “Elementary Programming Concepts”, and a key “Description” with a value “Fundamental concepts and skills of programming in a high-level language...” etc. Since key-value pairs are just plain texts, we incorporate key into the value text and use bi-GRU to encode the resulting text. Then the encoded texts for external domain knowledge are augmented into the memory. Apart from typical memory networks (Sukhbaatar, Weston, and Fergus 2015; Chen et al. 2016; Liu and Perez 2017) to encode dialog history and knowledge base (Madotto, Wu, and Fung 2018), our model enables self-attention among the utterance contents in memory. Typical memory networks only focus on enhancing the encoding of the last utterance in a dialog through the attention mechanism. However, the last utterance in a dialog history may not carry enough information for inference. Co-references among utterances in dialog history and external domain knowledge exists and can be modeled by self-attention. In our proposed model, we first perform self-attention over utterances in a dialog history. Resulting utterances are passed through an external domain knowledge modeled as another memory. Inspired by the gated memory network (Liu and Perez 2017), we employ the gating mechanism to control the degree of modification of utterance encodings in a multi-hop manner. We only put factual information, i.e. dialog utterances and external domain knowledge, into memory. Each response candidate is passed through the proposed GSMN trying to “retrieve” the factual information. Finally, we compute the summation of encodings of all utterances in a dialog history as a vector representation. Then we compute similarity between dialog history and each response candidate using a bilinear transform, followed by Softmax to obtain a probability distribution over response candidates.

Related Work

End-to-end neural network models for retrieval-based dialogue systems have been gaining popularity recently. One

[*] This work was done when Shuo Sun, Jie Cao and Zuohui Fu were interns at WeChat AI - Pattern Recognition Center, Tencent Inc.

[†] Equal contribution.

early work on dialog response selection employed a dual LSTM (Lowe et al. 2015b) on the Ubuntu corpus based on single-turn question-response pairs. In DSTC7 challenge, not only the dialogs have multiple turns, external knowledge is also provided requiring deeper level of understanding between contextual utterances and external knowledge. (Tan et al. 2015) built a bi-directional LSTM encoder with CNN on questions and responses candidates. Regarding multi-turn retrieval-based dialog system, (Wu et al. 2016) introduced a sequential matching method to distill important information between each contextual utterance and response pairs. (Lowe et al. 2015a) concatenated all the utterances in the passage and then matching score was computed based on the contextual representation. (Yan, Song, and Wu 2016) proposed contextual query reformulation strategies to concatenate contextual utterances with the last utterance. (Zhou et al. 2016) used a multi-view approach to model the contextual utterances as word sequence and utterance sequence. (Zhou et al. 2018) proposed self-attention and cross-attention in multi-turn response selection. Hierarchical encoding methods have received a lot of attention including web query suggestion (Sordoni et al. 2015), dialog systems (Serban et al. 2016; 2017; Bai et al. ; Tran, Zukerman, and Haffari 2017), and various document-level tasks (Li, Luong, and Jurafsky 2015; Tang, Qin, and Liu 2015; Yang et al. 2016).

For external knowledge integration, memory network (Sukhbaatar, Weston, and Fergus 2015) is a promising method for question and answering tasks with knowledge base. (Xiong, Merity, and Socher 2016) encoded knowledge into memory representation and retrieval is performed via the attention mechanism. (Chen et al. 2016) employed a memory network to store knowledges mentioned in dialog history for spoken language understanding. (Xu et al. 2016) incorporated a loosely-structured knowledge base into a neural network with the gating mechanism. Recently, (Yang et al. 2018) leverages online external knowledge for response ranking in information-seeking conversation systems.

Problem Statement

Given conversational history H, external knowledge G, question Q and response candidate pool $\{A_j\}$, we want to select the correct response candidate from the pool:

$$Pr(j'|H, G, Q, \{A_j\}) \tag{1}$$

Hierarchical LSTM Baseline

Inspired by hierarchical encoding method and attention mechanism, we investigate a hierarchical LSTM model as baseline. Encoding of dialog history is performed in a two-level hierarchy. Each utterance is first encoded using Bi-GRU. Motivated by match LSTM (Wang and Jiang 2016), we use the last utterance to "enhance" the rest of utterances using word-level attention. Then the first and last hidden vectors are concatenated to represent an utterance. Then a second-level Bi-GRU connects these utterance vectors followed by an utterance-level self-attention layer to capture the relationship among utterances.

Proposed System

Gated Self-attentive Memory Network

To tackle the problem on how to effectively incorporate external domain knowledge into the task of dialog response selection, we propose an End-to-end Gated Self-attentive Memory Network (GSMN) consisting of two sequential steps: retrieving relevant content from 1) the short-term memory and 2) the long-term memory. In this DSTC7 challenge, we define a short-term memory as utterances from a conversational history H, and a long-term memory as a set of entity subjects in external domain knowledge G=$\{g_1, g_2, g_3, ..., g_S\}$. We describe the building blocks of the proposed network in subsequent sections.

Embedding Layer We convert every word token t_i into word embedding e_i via an embedding lookup function ψ: $e_i = \psi(t_i)$.

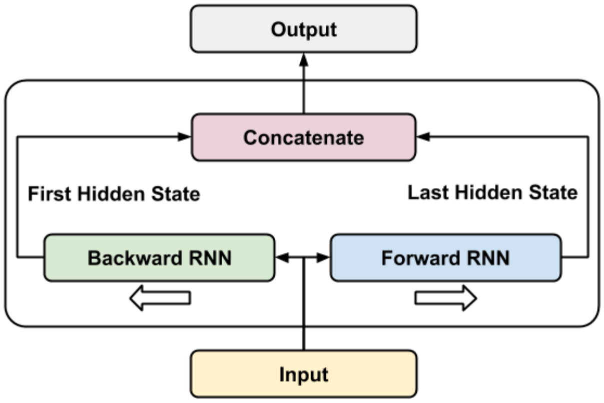


Figure 1: Bi-RNN based utterance encoding: The output vector is a concatenation of the first hidden state of a backward RNN and the last hidden state of a forward RNN.

Bi-RNN based Utterance Encoding We use bi-directional recurrent neural network (Bi-RNN) to encode each dialog utterance, external domain knowledge, and response candidate. In Figure 1, we use the concatenation of the last hidden state from a forward RNN and the first hidden state from a backward RNN to represent an utterance. Empirically, we experimented with Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) and Gated Recurrent Units (GRU)(Chung et al. 2014) and found that bi-GRU consistently outperformed bi-LSTM in our experiments.

Memory Attention Similar to memory network, we employ an attention mechanism to "retrieve" relevant memory vectors based on an input query vector. The output vector is a weighted sum of the memory vectors. In matrix form, we have multiple inputs $X = [x_1 \ ... \ x_J]$ and memory vectors $M = [m_1 \ ... \ m_K]$, $X \in \mathbb{R}^{J \times D}$, $M \in \mathbb{R}^{K \times D}$. The output matrix O serves as retrieved content and has the same dimension as the input matrix:

$$O = \text{Softmax}(X\phi_1(M)^T)\phi_2(M) \tag{2}$$

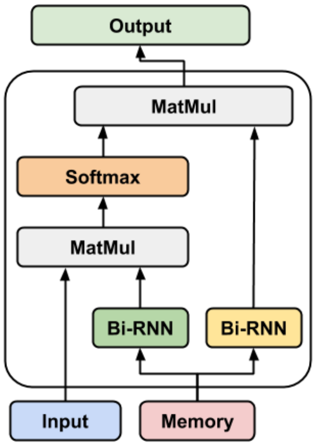


Figure 2: Memory attention module.

where ϕ_1 and ϕ_2 are Bi-RNN encoders with different trainable weights. Softmax is performed to a generate probability distributions over the memory items. Figure 2 shows the computation graph of memory attention mechanism.

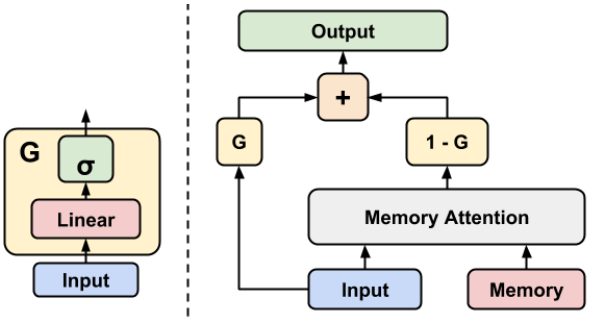


Figure 3: Gated memory attention module.

Gated Memory Attention Inspired by (Srivastava, Greff, and Schmidhuber 2015), we employ a gating mechanism when combining the input and the retrieved output from memory. The gating mechanism regulates the degree of enhancement of the input to prevent information overload. As shown in Figure 3, the gate G is a trainable fully connected neural network with sigmoid activation. The gating mechanism is shown below:

$$O' = G \cdot X + (1 - G) \cdot O \tag{3}$$

End-to-end GSMN Figure 4 depicts the end-to-end GSMN for dialog response selection. Short-term memories and long-term memories are stacked sequentially to enhance dialog utterances and response candidates. This process is repeated in a multi-hop fashion. Unlike (Sukhbaatar, Weston, and Fergus 2015), we do not share weights across hops. A reduce-sum operation converts the set of memory-enhanced dialog utterances into a single vector c to represent a full dialog history. Since each utterance vector attends to all utterances in a dialog history, GSMN is self-attentive by nature similar to transformer (Vaswani et al. 2017).

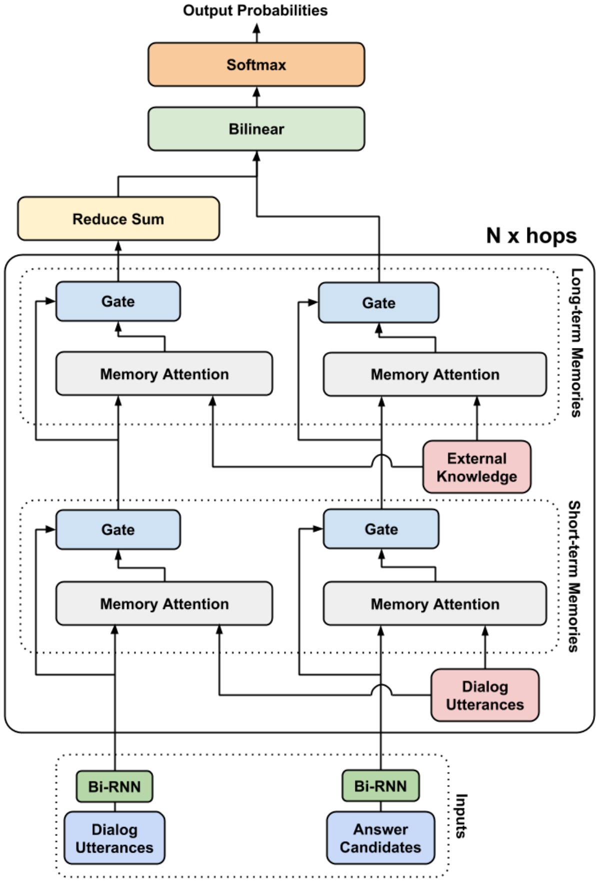


Figure 4: Our proposed end-to-end Gated Self-attentive Memory Network (GSMN) for dialog response selection. Gates and memory attention blocks in the same dotted box share parameters.

External Knowledge Encoding

Inspired by (Miller et al. 2016; Eric and Manning 2017; Madotto, Wu, and Fung 2018), we represent each key-value pair in external domain knowledge as a vector. In DSTC7 challenge, key and value are simply sequences of word tokens ($\{k_1, k_2, ..., k_M\}$, $\{v_1, v_2, ..., v_N\}$), where k_i is the i-th token in a key and v_i is the i-th token in a value. The final vector representation of a key-value pair is the concatenation of their average word embeddings $[\sum_{i=1}^{M} \psi(k_i), \sum_{i=1}^{N} \psi(v_i)]$. We treat each entity subject such as a course entity as a sequence of key-value vectors. We further compress the sequence via Bi-RNN taking the first and the last hidden vectors for representation. Finally, vectors from all entity subjects in external domain knowledge form a long-term memory in GSMN.

Dialog History and Response Relevance

Given the final encoding for the dialog history c and the encoding for an response candidate a_j, we measure their similarity as follows:

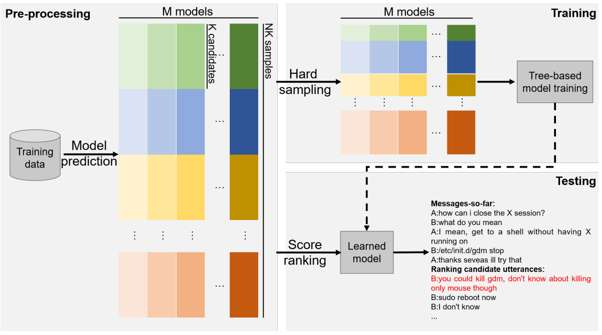
$$sim(c, a_j) = c^t \cdot M \cdot a_j \quad (4)$$

$$Pr(j|c) \propto e^{sim(c,a_j)} \quad (5)$$

where M is a trainable bi-linear transform of dialog history and response candidate encodings. We use Softmax to convert similarity scores into probability distribution over response candidates. Cross-entropy loss is employed for optimization.

Ensemble Learning

Ensemble learning is a popular technique to boost final performance after combining prediction results from different models. We propose a tree-based approach to fully exploit the complementary strengths of models. Tree-based machine learning algorithms (Chen and Guestrin 2016; Ke et al. 2017) are widely used in many competitions due to their effectiveness in improving predictive performance. For response selection, we formulate ensemble learning as binary classification. Denote N as the number of dialogs, M as the number of candidate models to ensemble, and K as the size of answer pool for each dialog. There are NK training samples with an M-dimension score vector from the models. We use XGBoost (Chen and Guestrin 2016) to train a binary classifier using scores from many GSMN models that are optimal at various evaluation metrics, and with various hyperparameters. Our goal is to ensemble diversified models. The ensemble learning pipeline is shown in Figure 5. To boost performance and speed up the training process, we filter out training instances with mean prediction scores outside the range [0.001,0.95] to allow the ensemble model focusing on "hard" samples. During test, response candidates for each dialog are ranked according to the predicted scores of the ensemble model.



Messages-so-far:
A:how can i close the X session?
B:what do you mean
A:I mean, get to a shell without having X running on
B:/etc/init.d/gdm stop
A:thanks seveas ill try that
Ranking candidate utterances:
B:you could kill gdm, don't know about killing only mouse though
B:sudo reboot now
B:I don't know
...

Figure 5: Tree-based ensemble learning. Different colors and saturation represent different dialogs and models respectively.

Experiments

We participated in the subtask 1 and 5 in the noetic end-to-end response selection challenge in DSTC7. In subtask 1, the challenge was to pick the correct option from a pool of 100 response candidates without referring to external knowledge. Subtask 5 was similar to subtask 1, except that external knowledge can be exploited.

Dataset	Train	Dev	Test	Test-Case 2
Advising	100k	500	500	500
Ubuntu	100k	5000	1000	N/A

Table 1: Number of samples in advising and Ubuntu datasets.

Data

Two datasets were provided for the challenge:

- **Flex Advising Corpus** contains student-advisor dialogs from a university. Each dialog is from an advising session in which the role of an advisor is to guide a student with choosing suitable courses for an upcoming semester. External knowledge include full course catalog and student profiles. Paraphrases of sentences in dialog history and target responses are also provided for data augmentation.
- **Ubuntu Dialog Corpus** contains dialogs from the Ubuntu IRC. External knowledge are the Linux manual pages.

Table 1 shows the statistics of the datasets. There were two test cases in the advising corpus but only the test-case 2 was used for final evaluation and ranking.

Settings

We used spaCy[1] to tokenize utterances. For the advising dataset, we added regular expressions to standardize the representations of course IDs. For example, "EECS 370", "370" and "EECS370" were uniformly converted to "eecs370". For the Ubuntu 3.0 dataset, we added rules to normalize command arguments such as "–help", "-h" etc. For subtask 5 of the advising dataset, we also inserted a course title after each course ID token in dialog utterances.

We initialized the word embedding layer using GloVe[2] (Pennington, Socher, and Manning 2014). Empirically, 300-dimension word vectors trained on 840-billion tokens gave us the best performances on both datasets. For the advising dataset, we performed data augmentation by randomly replacing dialog utterances and candidate responses with paraphrases, yielding 500,000 dialogs. To alleviate OOV issues, we evaluated different strategies. For the advising dataset, we used averaged word vectors of a course description to represent a course ID. For the Ubuntu dataset, we pre-trained another set of 300-dimension GloVe word vectors using the Ubuntu dataset. Then we combined the domain vectors to the off-the-shelf pre-trained GloVe vectors via summation. All word embeddings were kept fixed during GSMN training to prevent overfitting. We also experimented character-level embeddings but they did not help on top of the above strategies. We used Adam (Kingma and Ba 2015) to optimize the cross-entropy loss. The initial

[1] https://spacy.io

[2] `https://nlp.stanford.edu/projects/glove`

Model	R@1	R@10	R@50	MRR
Dual-LSTM baseline	0.062	0.296	0.728	N/A
HGRU baseline	0.164	0.632	0.922	0.299
SMN w/ 1 hop	0.218	0.642	0.956	0.337
2 hops	0.198	0.620	0.938	0.320
3 hops	0.206	0.648	0.942	0.333
GSMN w/ 1 hop	0.220	0.632	0.954	0.343
2 hops	0.214	0.644	**0.960**	0.338
3 hops	0.214	0.628	0.956	0.335
1 hop + EK	0.220	0.644	0.956	0.343
2 hops + EK	**0.224**	**0.654**	0.944	**0.354**

Table 2: Baseline and GSMN results on the Flex advising dev set. EK denotes external knowledge.

Model	R@1	R@10	R@50	MRR
Dual-LSTM baseline	0.083	0.360	0.804	N/A
SMN w/ 1 hop	0.326	0.671	0.952	0.445
SMN w/ 2 hop	0.337	0.686	0.956	0.455
GSMN w/ 1 hop	0.379	0.733	0.973	0.497
2 hops	0.389	**0.755**	0.972	0.508
3 hops	**0.398**	0.761	**0.976**	**0.515**

Table 3: Baseline and GSMN results on the Ubuntu dev set.

Measure	Ubuntu	Advising - Case 1	Advising - Case 2
Recall@1	0.475	0.494	0.18
Recall@10	0.814	0.85	0.562
Recall@50	0.978	0.98	0.94
MRR	0.595	0.6078	0.3069

Table 4: Official evaluation results on subtask 1 on Ubuntu and advising test sets.

learning rate was set to 1e-4 and the batch size was fixed at either 16 or 32 depending on computation resources. We applied dropout factor 0.3 on all modeling layers including word embedding to alleviate overfitting. We implemented our models using TensorFlow (Abadi et al. 2015) and conducted trainings on Nvidia GTI 1080TI GPUs. Each model were trained on a single GPU for few days. GSMN training took around 1-2 days to converge on the advising dataset and 4-5 days on the Ubuntu dataset. We trained different GSMN models by varying the number of hops from one to three and the type of bi-RNN encoder either using LSTM or GRU. The number of selected models for ensembling was around 20. For subtask 1, we trained GSMN models without external knowledge and thus excluded the long-term memories. Our final results were produced via ensembling many GSMN models and hierarchical-GRU baselines.

Experimental Results

Table 2 shows results using various models. Hierarchical GRU baseline yielded substantial improvement compared to the dual LSTM baseline provided by DSTC7, showing that word-level attention and hierarchical encoding of dialog history helped. In addition, GSMN outperformed hierarchical GRU on all metrics. The best number of hops was not consistent on various metrics. Incorporating external knowledge into GSMN generally helped. Table 3 shows results on the Ubuntu dev set. We obtained similar trend on model performance. The gating mechanism helped more on Ubuntu than advising dev set. Unfortunately, we were unable to fully explore the effect of using long-term memories on Ubuntu dataset due to time constraints. Table 4-5 present our official evaluation results. Our proposed system after ensembling ranked at the second place on advising and Ubuntu subtask 5 using external knowledge. Simple averaging of system ensembles degraded the overall evaluation metric by around 1 absolute point.

Measure	Ubuntu	Advising - Case 1	Advising - Case 2
Recall@1	0.504	0.538	0.178
Recall@10	0.827	0.864*	0.608
Recall@50	0.98	0.986	0.944
MRR	0.6172	0.6455*	0.3149

Table 5: Official evaluation results on subtask 5 (with external knowledge) on Ubuntu and advising test sets. * denotes the 1st-place evaluation results.

Discussion

Inspired by machine reading literature, hierarchical GRU using word-level attention was much better than the dual LSTM baseline. On the other hand, GSMN with utterance-level self-attention turned out to be more effective and outperformed hierarchical GRU. To the best of our knowledge, most previous techniques for dialog response selection tasks focused on enhancing the encoding of the last utterance in a dialog through the attention mechanism. However, the last utterance in a dialog history may not carry enough information for inference. Co-references among utterances in a dialog history and external domain knowledge exist and can be modeled by self-attention, which is the key component of GSMN.

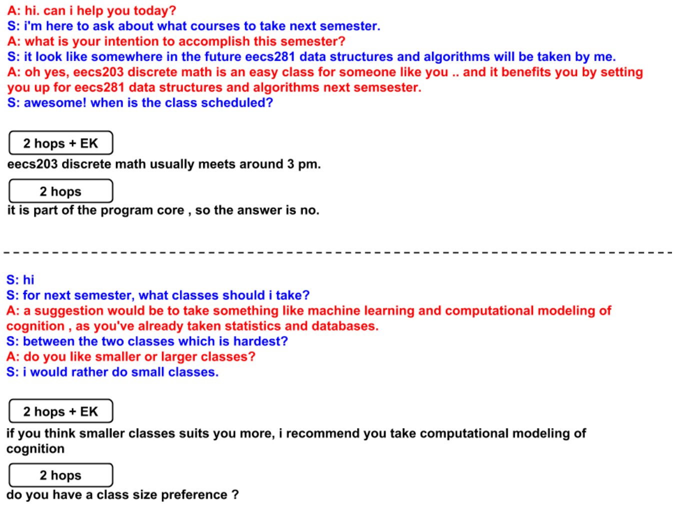
To further explore the effectiveness of using self-attention at utterance level, we trained and evaluated various GSMN models but restricted the inputs to only the last utterance of a dialog history. Consequently, only the last utterances were enhanced by the memory networks and used to compute similarity scores with response candidates. In Table 6, we observed significant performance gains across all evaluation metrics and model settings, confirming that modeling co-references among dialog utterances via self-attention was effective.

We also hypothesized that the long-term memories of GSMN helped picking the most relevant response candidates. We observed that many examples using a 2-hops GSMN picked the correct response candidates successfully with the help of external knowledge. In the first example in Figure 6, the model correctly identified the fact that the

Model	R@1			R@5			R@10			MRR		
	LU	AU	Gain	LU	AU	Gain	LU	AU	Gain	LU	AU	Gain
1 hop	0.194	0.220	▲0.026	0.592	0.643	▲0.051	0.910	0.954	▲0.044	0.317	0.343	▲0.026
2 hops	0.208	0.214	▲0.006	0.596	0.644	▲0.048	0.920	0.960	▲0.040	0.326	0.338	▲0.012
3 hops	0.208	0.214	▲0.006	0.616	0.628	▲0.012	0.928	0.956	▲0.028	0.329	0.335	▲0.006
1 hop + EK	0.172	0.220	▲0.048	0.578	0.644	▲0.066	0.910	0.956	▲0.046	0.294	0.343	▲0.049
2 hops + EK	0.186	0.224	▲0.038	0.602	0.654	▲0.052	0.932	0.944	▲0.012	0.319	0.354	▲0.035

Table 6: Experiment results on advising dev set using only last utterance (LU) of every dialog as compared to using all utterances (AU) of every dialog as inputs to Gated Self-attentive Memory Networks. EK denotes external knowledge.

student's query was about the schedule of "eecs203" that required retrieving the class time from the course catalog. In the second example, the model could only figure out that "computational modeling of cognition" is the smaller class with the help of external knowledge. Therefore, GSMN is promising to incorporate external knowledge into dialog history encoding. Although our model achieved the highest scores among the participating teams on the advising test-case 1 in subtask 5, it did not perform as well on test-case 2. We inspected some examples in test-case 2 and realized that our model made mistakes on "easy" examples. We discovered that our model could not handle cases where the subject and direct object of sentence were reversed, such as "EECS281 is taken by most students in second semester" instead of "Most students take EECS281 in second semester". We suspect that utterance encoding using recurrent neural networks in GSMN might have heavily memorized the language styles in the training data and failed to adapt to the novel styles in test-case 2.



A: hi. can i help you today?
S: i'm here to ask about what courses to take next semester.
A: what is your intention to accomplish this semester?
S: it look like somewhere in the future eecs281 data structures and algorithms will be taken by me.
A: oh yes, eecs203 discrete math is an easy class for someone like you .. and it benefits you by setting you up for eecs281 data structures and algorithms next semsester.
S: awesome! when is the class scheduled?

2 hops + EK
eecs203 discrete math usually meets around 3 pm.

2 hops
it is part of the program core , so the answer is no.

S: hi
S: for next semester, what classes should i take?
A: a suggestion would be to take something like machine learning and computational modeling of cognition , as you've already taken statistics and databases.
S: between the two classes which is hardest?
A: do you like smaller or larger classes?
S: i would rather do small classes.

2 hops + EK
if you think smaller classes suits you more, i recommend you take computational modeling of cognition

2 hops
do you have a class size preference ?

Figure 6: Sample dialogs from advising dev set which were correctly answered by GSMN with the help of external knowledge.

Conclusions

We have presented Gated Self-attentive Memory Network for dialog response selection. Our proposed approach models dialog history and external knowledge as short-term and long-term memories respectively. We encode each subject entity in external domain knowledge as a sequence of key-value pairs with pre-trained embeddings. Experimental results have shown that Gated Self-attentive Memory Network effectively integrates external knowledge and dialog history in an end-to-end fashion. We achieve the second place in the subtask 5 of the DSTC7 response selection challenge. For future work, we believe that improving the encoding power of dialog history and external domain knowledge as well as their interaction will be crucial for further performance improvement.

References

Bai, Z.; Yu, B.; Chen, G.; Wang, B.; and Wang, Z. Modeling conversations to learn responding policies of e2e task-oriented dialog system.

Chen, T., and Guestrin, C. 2016. XGBoost: A scalable tree boosting system. In *22nd SIGKDD Conference on Knowledge Discovery and Data Mining*.

Chen, Y.-N.; Hakkani-Tur, D.; Tur, G.; Gao, J.; and Deng, L. 2016. End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS Deep Learning and Representation Learning Workshop*.

Eric, M., and Manning, C. 2017. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 468–473. Association for Computational Linguistics.

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

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems (NIPS)*.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Li, J.; Luong, M.-T.; and Jurafsky, D. 2015. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*.

Liu, F., and Perez, J. 2017. Gated end-to-end memory networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Lowe, R.; Pow, N.; Serban, I.; Charlin, L.; and Pineau, J. 2015a. Incorporating unstructured textual knowledge sources into neural dialogue systems. In *Neural Information Processing Systems Workshop on Machine Learning for Spoken Language Understanding*.

Lowe, R.; Pow, N.; Serban, I. V.; and Pineau, J. 2015b. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of SIGDIAL*.

Madotto, A.; Wu, C.-S.; and Fung, P. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 1468–1478. Association for Computational Linguistics.

Martín, A. et. al. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems.

Miller, A.; Fisch, A.; Dodge, J.; Karimi, A.-H.; Bordes, A.; and Weston, J. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1400–1409. Association for Computational Linguistics.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A. C.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, volume 16, 3776–3784.

Serban, I. V.; Klinger, T.; Tesauro, G.; Talamadupula, K.; Zhou, B.; Bengio, Y.; and Courville, A. C. 2017. Multiresolution recurrent neural networks: An application to dialogue response generation. In *AAAI*, 3288–3294.

Sordoni, A.; Bengio, Y.; Vahabi, H.; Lioma, C.; Grue Simonsen, J.; and Nie, J.-Y. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 553–562. ACM.

Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

Sukhbaatar, S.; Weston, J.; and Fergus, R. 2015. End-to-end memory networks. In *Advances in neural information processing systems*.

Tan, M.; Santos, C. d.; Xiang, B.; and Zhou, B. 2015. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

Tang, D.; Qin, B.; and Liu, T. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 1422–1432.

Tran, Q. H.; Zukerman, I.; and Haffari, G. 2017. A hierarchical neural model for learning sequences of dialogue acts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, 428–437.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*.

Wang, S., and Jiang, J. 2016. Machine comprehension using match-lstm and answer pointer.

Wu, Y.; Wu, W.; Xing, C.; Zhou, M.; and Li, Z. 2016. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. *arXiv preprint arXiv:1612.01627*.

Xiong, C.; Merity, S.; and Socher, R. 2016. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, 2397–2406.

Xu, Z.; Liu, B.; Wang, B.; Sun, C.; and Wang, X. 2016. Incorporating loose-structured knowledge into lstm with recall gate for conversation modeling. *arXiv preprint arXiv:1605.05110*.

Yan, R.; Song, Y.; and Wu, H. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 55–64. ACM.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1480–1489.

Yang, L.; Qiu, M.; Qu, C.; Guo, J.; Zhang, Y.; Croft, W. B.; Huang, J.; and Chen, H. 2018. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. *arXiv preprint arXiv:1805.00188*.

Yoshino, K.; Hori, C.; Perez, J.; D'Haro, L. F.; Polymenakos, L.; Gunasekara, C.; Lasecki, W. S.; Kummerfeld, J.; Galley, M.; Brockett, C.; Gao, J.; Dolan, B.; Gao, S.; Marks, T. K.; Parikh, D.; and Batra, D. 2018. The 7th dialog system technology challenge. *arXiv preprint*.

Zhou, X.; Dong, D.; Wu, H.; Zhao, S.; Yu, D.; Tian, H.; Liu, X.; and Yan, R. 2016. Multi-view response selection for human-computer conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 372–381.

Zhou, X.; Li, L.; Dong, D.; Liu, Y.; Chen, Y.; Zhao, W. X.; Yu, D.; and Wu, H. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of ACL*.