Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems

Andrea Madotto*, **Chien-Sheng Wu***, **Pascale Fung**
Human Language Technology Center
Center for Artificial Intelligence Research (CAiRE)
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
[eeandreamad,cwuak,pascale]@ust.hk

Abstract

End-to-end task-oriented dialog systems usually suffer from the challenge of incorporating knowledge bases. In this paper, we propose a novel yet simple end-to-end differentiable model called memory-to-sequence (Mem2Seq) to address this issue. Mem2Seq is the first neural generative model that combines the multi-hop attention over memories with the idea of pointer network. We empirically show how Mem2Seq controls each generation step, and how its multi-hop attention mechanism helps in learning correlations between memories. In addition, our model is quite general without complicated task-specific designs. As a result, we show that Mem2Seq can be trained faster and attain the state-of-the-art performance on three different task-oriented dialog datasets.

1 Introduction

Task-oriented dialog systems help users to achieve specific goals with natural language such as restaurant reservation and schedule arrangement. Traditionally, they have been built with several pipelined modules: language understanding, dialog management, knowledge query, and language generation (Williams and Young, 2007; Hori et al., 2009; Lee et al., 2009; Levin et al., 2000; Young et al., 2013). Moreover, the ability to query external Knowledge Bases (KBs) is essential in task-oriented dialog systems, since the responses are guided not only by the dialog history but also by the query results (e.g. Table 1). However, despite the stability of such pipelined systems via combining domain-specific knowledge and slot-filling techniques, modeling the dependencies between modules is complex and the KB interpretation requires human effort.

Point of interest (poi)	Distance	Traffic info	Poi type	Address
The Westin	5 miles	moderate traffic	rest stop	329 El Camino Real
Round Table	4 miles	no traffic	pizza restaurant	113 Anton Ct
Mandarin Roots	5 miles	no traffic	chinese restaurant	271 Springer Street
Palo Alto Cafe	4 miles	moderate traffic	coffee or tea place	436 Alger Dr
Dominos	6 miles	heavy traffic	pizza restaurant	776 Arastradero Rd
Stanford Express Care	6 miles	no traffic	hospital	214 El Camino Real
Hotel Keen	2 miles	heavy traffic	rest stop	578 Arbol Dr

1th Turn	**DRIVER**	Where can I get tea?
	Seq2Seq	I have a away from away would you like the address
	+Attn	I have a listing for a place that serves tea that is 5 miles away
	Ptr-Unk	There is a away you like would you like more info
	Mem2Seq	The nearest tea shop is Palo Alto Cafe located 4 miles away, would you like directions there?
	GOLD	Palo Alto Cafe is 4 miles away and serves coffee and tea. Do you want the address?
2th Turn	**DRIVER**	Yes.
	Seq2Seq	The is at at
	+Attn	The address is 329 El Camino Real and it's 3 miles away and there is no traffic
	Ptr-Unk	Palo Alto Cafe is 4 miles away and PAD is no traffic
	Mem2Seq	Palo Alto Cafe is 4 miles away at 436 Alger Drive
	GOLD	Palo Alto is located at 436 Alger Dr.

Table 1: Example of generated responses for the In-Car Assistant on the navigation domain.

Recently, end-to-end approaches for dialog modeling, which use recurrent neural networks (RNN) encoder-decoder models, have shown promising results (Serban et al., 2016; Wen et al., 2017; Zhao et al., 2017). Since they can directly map plain text dialog history to the output responses, and the dialog states are latent, there is no need for hand-crafted state labels. Moreover, attention-based copy mechanism (Gulcehre et al., 2016; Eric and Manning, 2017) have been recently introduced to copy words directly from the input sources to the output responses. Using such mechanism, even when unknown tokens appear in the dialog history, the models are still able to produce correct and relevant entities.

However, although the above mentioned approaches were successful, they still suffer from two main problems: 1) They struggle to effectively incorporate external KB information into the RNN hidden states (Sukhbaatar et al., 2015),

* These two authors contributed equally.

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 1468–1478
Melbourne, Australia, July 15 - 20, 2018.

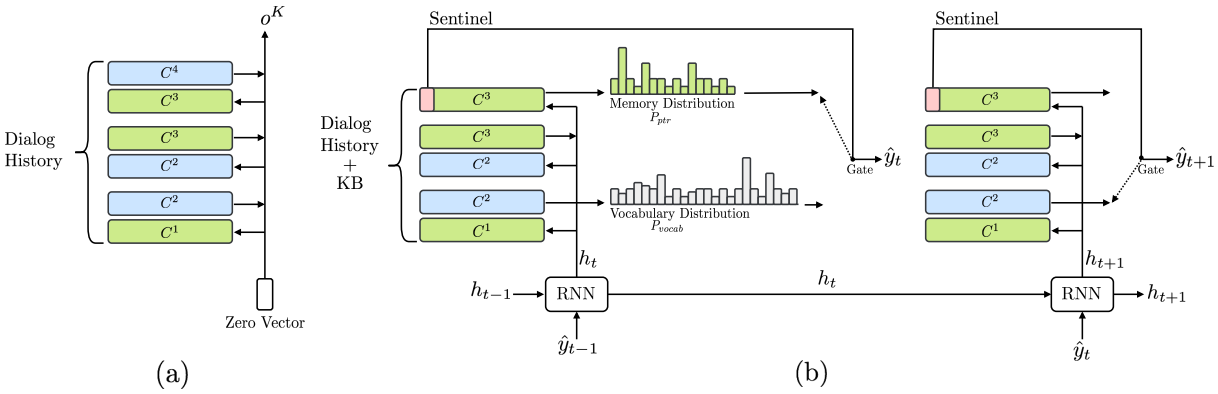

Figure 1: The proposed Mem2Seq architecture for task-oriented dialog systems. (a) Memory encoder with 3 hops; (b) Memory decoder over 2 step generation.

since RNNs are known to be unstable over long sequences. 2) Processing long sequences is very time-consuming, especially when using attention mechanisms.

On the other hand, end-to-end memory networks (MemNNs) are recurrent attention models over a possibly large external memory (Sukhbaatar et al., 2015). They write external memories into several embedding matrices, and use query vectors to read memories repeatedly. This approach can memorize external KB information and rapidly encode long dialog history. Moreover, the multi-hop mechanism of MemNN has empirically shown to be essential in achieving high performance on reasoning tasks (Bordes and Weston, 2017). Nevertheless, MemNN simply chooses its responses from a predefined candidate pool rather than generating word-by-word. In addition, the memory queries need explicit design rather than being learned, and the copy mechanism is absent.

To address these problems, we present a novel architecture that we call Memory-to-Sequence (Mem2Seq) to learn task-oriented dialogs in an end-to-end manner. In short, our model augments the existing MemNN framework with a sequential generative architecture, using global multi-hop attention mechanisms to copy words directly from dialog history or KBs. We summarize our main contributions as such: 1) Mem2Seq is the first model to combine multi-hop attention mechanisms with the idea of pointer networks, which allows us to effectively incorporate KB information. 2) Mem2Seq learns how to generate dynamic queries to control the memory access. In addition, we visualize and interpret the model dynamics among hops for both the memory controller and the attention. 3) Mem2Seq can be trained faster and achieve state-of-the-art results in several task-oriented dialog datasets.

2 Model Description

Mem2Seq [1] is composed of two components: the MemNN encoder, and the memory decoder as shown in Figure 1. The MemNN encoder creates a vector representation of the dialog history. Then the memory decoder reads and copies the memory to generate a response. We define all the words in the dialog history as a sequence of tokens $X = \{x_1, \ldots, x_n, \$\}$, where $\$$ is a special charter used as a sentinel, and the KB tuples as $B = \{b_1, \ldots, b_l\}$. We further define $U = [B; X]$ as the concatenation of the two sets X and B, $Y = \{y_1, \ldots, y_m\}$ as the set of words in the expected system response, and $PTR = \{ptr_1, \ldots, ptr_m\}$ as the pointer index set:

$$ptr_i = \begin{cases} max(z) & \text{if } \exists z \text{ s.t. } y_i = u_z \\ n + l + 1 & \text{otherwise} \end{cases} \quad (1)$$

where $u_z \in U$ is the input sequence and $n + l + 1$ is the sentinel position index.

2.1 Memory Encoder

Mem2Seq uses a standard MemNN with adjacent weighted tying (Sukhbaatar et al., 2015) as an encoder. The input of the encoder is word-level information in U. The memories of MemNN are represented by a set of trainable embedding matrices $C = \{C^1, \ldots, C^{K+1}\}$, where each C^k maps tokens to vectors, and a query vector q^k is used as a reading head. The model loops over K hops and

[1] The code is available at https://github.com/HLTCHKUST/Mem2Seq

it computes the attention weights at hop k for each memory i using:

$$p_i^k = \text{Softmax}((q^k)^T C_i^k), \quad (2)$$

where $C_i^k = C^k(x_i)$ is the memory content in position i, and $\text{Softmax}(z_i) = e^{z_i}/\Sigma_j e^{z_j}$. Here, p^k is a soft memory selector that decides the memory relevance with respect to the query vector q^k. Then, the model reads out the memory o^k by the weighted sum over C^{k+1} [2],

$$o^k = \sum_i p_i^k C_i^{k+1}. \quad (3)$$

Then, the query vector is updated for the next hop by using $q^{k+1} = q^k + o^k$. The result from the encoding step is the memory vector o^K, which will become the input for the decoding step.

2.2 Memory Decoder

The decoder uses RNN and MemNN. The MemNN is loaded with both X and B, since we use both dialog history and KB information to generate a proper system response. A Gated Recurrent Unit (GRU) (Chung et al., 2014), is used as a dynamic query generator for the MemNN. At each decoding step t, the GRU gets the previously generated word and the previous query as input, and it generates the new query vector. Formally:

$$h_t = \text{GRU}(C^1(\hat{y}_{t-1}), h_{t-1}); \quad (4)$$

Then the query h_t is passed to the MemNN which will produce the token, where h_0 is the encoder vector o^K. At each time step, two distribution are generated: one over all the words in the vocabulary (P_{vocab}), and one over the memory contents (P_{ptr}), which are the dialog history and KB inofrmation. The first, P_{vocab}, is generated by concatenating the first hop attention read out and the current query vector.

$$P_{vocab}(\hat{y}_t) = \text{Softmax}(W_1[h_t; o^1]) \quad (5)$$

where W_1 is a trainable parameter. On the other hand, P_{ptr} is generated using the attention weights at the last MemNN hop of the decoder: $P_{ptr} = p_t^K$. Our decoder generates tokens by pointing to the input words in the memory, which is a similar mechanism to the attention used in pointer networks (Vinyals et al., 2015).

[2] Here is C^{k+1} since we use adjacent weighted tying.

We designed our architecture in this way because we expect the attention weights in the first and the last hop to show a "looser" and "sharper" distribution, respectively. To elaborate, the first hop focuses more on retrieving memory information and the last one tends to choose the exact token leveraging the pointer supervision. Hence, during training all the parameters are jointly learned by minimizing the sum of two standard cross-entropy losses: one between $P_{vocab}(\hat{y}_t)$ and $y_t \in Y$ for the vocabulary distribution, and one between $P_{ptr}(\hat{y}_t)$ and $ptr_t \in PTR$ for the memory distribution.

2.2.1 Sentinel

If the expected word is not appearing in the memories, then the P_{ptr} is trained to produce the sentinel token \$, as shown in Equation 1. Once the sentinel is chosen, our model generates the token from P_{vocab}, otherwise, it takes the memory content using the P_{ptr} distribution. Basically, the sentinel token is used as a hard gate to control which distribution to use at each time step. A similar approach has been used in (Merity et al., 2017) to control a soft gate in a language modeling task. With this method, the model does not need to learn a gating function separately as in Gulcehre et al. (2016), and is not constrained by a soft gate function as in See et al. (2017).

2.3 Memory Content

We store word-level content X in the memory module. Similar to Bordes and Weston (2017), we add temporal information and speaker information in each token of X to capture the sequential dependencies. For example, "hello $t1$ \$$u$" means "hello" at time step 1 spoken by a user.

On the other hand, to store B, the KB information, we follow the works of Miller et al. (2016); Eric et al. (2017) that use a (subject, relation, object) representation. For example, we represent the information of The Westin in Table 1: (The Westin, Distance, 5 miles). Thus, we sum word embeddings of the subject, relation, and object to obtain each KB memory representation. During decoding stage, the object part is used as the generated word for P_{ptr}. For instance, when the KB tuple (The Westin, Distance, 5 miles) is pointed, our model copies "5 miles" as an output word. Notice that only a specific section of the KB, relevant to a specific dialog, is loaded into the memory.

Task	1	2	3	4	5	DSTC2	In-Car
Avg. User turns	4	6.5	6.4	3.5	12.9	6.7	2.6
Avg. Sys turns	6	9.5	9.9	3.5	18.4	9.3	2.6
Avg. KB results	0	0	24	7	23.7	39.5	66.1
Avg. Sys words	6.3	6.2	7.2	5.7	6.5	10.2	8.6
Max. Sys words	9	9	9	8	9	29	87
Pointer Ratio	.23	.53	.46	.19	.60	.46	.42
Vocabulary	3747					1229	1601
Train dialogs	1000					1618	2425
Val dialogs	1000					500	302
Test dialogs	1000 + 1000 OOV					1117	304

Table 2: Dataset statistics for 3 different datasets.

3 Experimental Setup

3.1 Dataset

We use three public multi-turn task-oriented dialog datasets to evaluate our model: the bAbI dialog (Bordes and Weston, 2017), DSTC2 (Henderson et al., 2014) and In-Car Assistant (Eric et al., 2017). The train/validation/test sets of these three datasets are split in advance by the providers. The dataset statistics are reported in Table 2.

The bAbI dialog includes five end-to-end dialog learning tasks in the restaurant domain, which are simulated dialog data. Task 1 to 4 are about API calls, refining API calls, recommending options, and providing additional information, respectively. Task 5 is the union of tasks 1-4. There are two test sets for each task: one follows the same distribution as the training set and the other has out-of-vocabulary (OOV) entity values that does not exist in the training set.

We also used dialogs extracted from the Dialog State Tracking Challenge 2 (DSTC2) with the refined version from Bordes and Weston (2017), which ignores the dialog state annotations. The main difference with bAbI dialog is that this dataset is extracted from real human-bot dialogs, which is noisier and harder since the bots made mistakes due to speech recognition errors or misinterpretations.

Recently, In-Car Assistant dataset has been released. which is a human-human, multi-domain dialog dataset collected from Amazon Mechanical Turk. It has three distinct domains: calendar scheduling, weather information retrieval, and point-of-interest navigation. This dataset has shorter conversation turns, but the user and system behaviors are more diverse. In addition, the system responses are variant and the KB information is much more complicated. Hence, this dataset requires stronger ability to interact with KBs, rather than dialog state tracking.

3.2 Training

We trained our model end-to-end using Adam optimizer (Kingma and Ba, 2015), and chose learning rate between $[1e^{-3}, 1e^{-4}]$. The MemNNs, both encoder and decoder, have hops $K = 1, 3, 6$ to show the performance difference. We use simple greedy search and without any re-scoring techniques. The embedding size, which is also equivalent to the memory size and the RNN hidden size (i.e., including the baselines), has been selected between $[64, 512]$. The dropout rate is set between $[0.1, 0.4]$, and we also randomly mask some input words into unknown tokens to simulate OOV situation with the same dropout ratio. In all the datasets, we tuned the hyper-parameters with grid-search over the validation set, using as measure to the Per-response Accuracy for bAbI dialog and DSTC2, and BLEU score for the In-Car Assistant.

3.3 Evaluation Metrics

Per-response/dialog Accuracy: A generative response is correct only if it is exactly the same as the gold response. A dialog is correct only if every generated responses of the dialog are correct, which can be considered as the task-completion rate. Note that Bordes and Weston (2017) tests their model by selecting the system response from predefined response candidates, that is, their system solves a multi-class classification task. Since Mem2Seq generates each token individually, evaluating with this metric is much more challenging for our model.

BLEU: It is a measure commonly used for machine translation systems (Papineni et al., 2002), but it has also been used in evaluating dialog systems (Eric and Manning, 2017; Zhao et al., 2017) and chat-bots (Ritter et al., 2011; Li et al., 2016). Moreover, BLEU score is a relevant measure in task-oriented dialog as there is not a large variance between the generated answers, unlike open domain generation (Liu et al., 2016). Hence, we include BLEU score in our evaluation (i.e. using Moses `multi-bleu.perl` script).

Entity F1: We micro-average over the entire set of system responses and compare the entities in plain text. The entities in each gold system response are selected by a predefined entity list. This metric evaluates the ability to generate relevant entities from the provided KBs and to capture the semantics of the dialog (Eric and Manning, 2017; Eric et al., 2017). Note that the original In-Car Assis-

Task	QRN	MemNN	GMemNN	Seq2Seq	Seq2Seq+Attn	Ptr-Unk	Mem2Seq H1	Mem2Seq H3	Mem2Seq H6
T1	99.4 (-)	99.9 (99.6)	100 (100)	100 (100)	100 (100)	100 (100)	100 (100)	100 (100)	100 (100)
T2	99.5 (-)	100 (100)	100 (100)	100 (100)	100 (100)	100 (100)	100 (100)	100 (100)	100 (100)
T3	74.8 (-)	74.9 (2.0)	74.9 (0)	74.8 (0)	74.8 (0)	85.1 (19.0)	87.0 (25.2)	94.5 (59.6)	**94.7** (**62.1**)
T4	57.2 (-)	59.5 (3.0)	57.2 (0)	57.2 (0)	57.2 (0)	100 (100)	97.6 (91.7)	100 (100)	**100** (**100**)
T5	**99.6** (-)	96.1 (49.4)	96.3 (52.5)	98.8 (81.5)	98.4 (87.3)	99.4 (91.5)	96.1 (45.3)	98.2 (72.9)	97.9 (69.6)
T1-OOV	83.1 (-)	72.3 (0)	82.4 (0)	79.9 (0)	81.7 (0)	92.5 (54.7)	93.4 (60.4)	91.3 (52.0)	**94.0** (**62.2**)
T2-OOV	78.9 (-)	78.9 (0)	78.9 (0)	78.9 (0)	78.9 (0)	83.2 (0)	81.7 (1.2)	84.7 (7.3)	**86.5** (**12.4**)
T3-OOV	75.2 (-)	74.4 (0)	75.3 (0)	74.3 (0)	75.3 (0)	82.9 (13.4)	86.6 (26.2)	**93.2** (**53.3**)	90.3 (38.7)
T4-OOV	56.9 (-)	57.6 (0)	57.0 (0)	57.0 (0)	57.0 (0)	100 (100)	97.3 (90.6)	100 (100)	**100** (**100**)
T5-OOV	67.8 (-)	65.5 (0)	66.7 (0)	67.4 (0)	65.7 (0)	73.6 (0)	67.6 (0)	78.1 (0.4)	**84.5** (**2.3**)

Table 3: Per-response and per-dialog (in the parentheses) accuracy on bAbI dialogs. Mem2Seq achieves the highest average per-response accuracy and has the least out-of-vocabulary performance drop.

	Ent. F1	BLEU	Per-Resp.	Per-Dial.
Rule-Based	-	-	33.3	-
QRN	-	-	43.8	-
MemNN	-	-	41.1	0.0
GMemNN	-	-	**47.4**	1.4
Seq2Seq	69.7	55.0	46.4	**1.5**
+Attn	67.1	**56.6**	46.0	1.4
+Copy	71.6	55.4	47.3	1.3
Mem2Seq H1	72.9	53.7	41.7	0.0
Mem2Seq H3	**75.3**	55.3	45.0	0.5
Mem2Seq H6	72.8	53.6	42.8	0.7

Table 4: Evaluation on DSTC2. Seq2Seq (+attn and +copy) is reported from Eric and Manning (2017).

	BLEU	Ent. F1	Sch. F1	Wea. F1	Nav. F1
*Human**	13.5	60.7	64.3	61.6	55.2
*Rule-Based**	6.6	43.8	61.3	39.5	40.4
*KV Retrieval Net**	13.2	48.0	62.9	47.0	41.3
Seq2Seq	8.4	10.3	09.7	14.1	07.0
+Attn	9.3	19.9	23.4	25.6	10.8
Ptr-Unk	8.3	22.7	26.9	26.7	14.9
Mem2Seq H1	11.6	32.4	39.8	**33.6**	**24.6**
Mem2Seq H3	**12.6**	**33.4**	**49.3**	32.8	20.0
Mem2Seq H6	9.9	23.6	34.3	33.0	4.4

Table 5: Evaluation on In-Car Assistant. Human, rule-based and KV Retrieval Net evaluation (with *) are reported from (Eric et al., 2017), which are not directly comparable. Mem2Seq achieves highest BLEU and entity F1 score over baselines.

tant F1 scores reported in Eric et al. (2017) uses the entities in their canonicalized forms, which are not calculated based on real entity value. Since the datasets are not designed for slot-tracking, we report entity F1 rather than the slot-tracking accuracy as in (Wen et al., 2017; Zhao et al., 2017).

4 Experimental Results

We mainly compare Mem2Seq with hop 1,3,6 with several existing models: query-reduction networks (QRN, Seo et al. (2017)), end-to-end memory networks (MemNN, Sukhbaatar et al. (2015)), and gated end-to-end memory networks (GMemNN, Liu and Perez (2017)). We also implemented the following baseline models: standard sequence-to-sequence (Seq2Seq) models with and without attention (Luong et al., 2015), and pointer to unknown (Ptr-Unk, Gulcehre et al. (2016)). Note that the results we listed in Table 3 and Table 4 for QRN are different from the original paper, because based on their released code, [3] we discovered that the per-response accuracy was not correctly computed.

bAbI Dialog: In Table 3, we follow Bordes and Weston (2017) to compare the performance based on per-response and per-dialog accuracy. Mem2Seq with 6 hops can achieve per-response 97.9% and per-dialog 69.6% accuracy in T5, and 84.5% and 2.3% for T5-OOV, which surpass existing methods by far. One can find that in T3 especially, which is the task to recommend restaurant based on their ranks, our model can achieve promising results due to the memory pointer. In terms of per-response accuracy, this indicates that our model can generalize well with few performance loss for test OOV data, while others have around 15-20% drop. The performance gain in OOV data is also mainly attributed to the use of copy mechanism. In addition, the effectiveness of hops is demonstrated in tasks 3-5, since they require reasoning ability over the KB information. Note that QRN, MemNN and GMemNN viewed bAbI dialog tasks as classification problems. Although their tasks are easier compared to our generative methods, Mem2Seq models can still overpass the performance. Finally, one can find that Seq2Seq and Ptr-Unk models are also strong baselines, which further confirms that generative methods can also achieve good performance in task-oriented dialog systems (Eric and Manning, 2017).

[3] We simply modified the evaluation part and reported the results. (https://github.com/uwnlp/qrn)

DSTC2: In Table 4, the Seq2Seq models from Eric and Manning (2017) and the rule-based from Bordes and Weston (2017) are reported. Mem2Seq has the highest 75.3% entity F1 score and an high of 55.3 BLEU score. This further confirms that Mem2Seq can perform well in retrieving the correct entity, using the multiple hop mechanism without losing language modeling. Here, we do not report the results using match type (Bordes and Weston, 2017) or entity type (Eric and Manning, 2017) feature, since this meta-information are not commonly available and we want to have an evaluation on plain input output couples. One can also find out that, Mem2Seq comparable per-response accuracy (i.e. 2% margin) among other existing solution. Note that the per-response accuracy for every model is less than 50% since the dataset is quite noisy and it is hard to generate a response that is exactly the same as the gold one.

In-Car Assistant: In Table 5, our model can achieve highest 12.6 BLEU score. In addition, Mem2Seq has shown promising results in terms of Entity F1 scores (33.4%), which are, in general, much higher than those of other baselines. Note that the numbers reported from Eric et al. (2017) are not directly comparable to ours as we mention below. The other baselines such as Seq2Seq or Ptr-Unk especially have worse performances in this dataset since it is very inefficient for RNN methods to encode longer KB information, which is the advantage of Mem2Seq.

Furthermore, we observe an interesting phenomenon that humans can easily achieve a high entity F1 score with a low BLEU score. This implies that stronger reasoning ability over entities (hops) is crucial, but the results may not be similar to the golden answer. We believe humans can produce good answers even with a low BLEU score, since there could be different ways to express the same concepts. Therefore, Mem2Seq shows the potential to successfully choose the correct entities.

Note that the results of KV Retrieval Net baseline reported in Table 5 come from the original paper (Eric et al., 2017) of In-Car Assistant, where they simplified the task by mapping the expression of entities to a canonical form using named entity recognition (NER) and linking. Hence the evaluation is not directly comparable to our system. For example, their model learned to generate responses such as "You have a football game at football_time with football_party," instead of generating a sentence such as "You have a football game at 7 pm with John." Since there could be more than one football_party or football_time, their model does not learn how to access the KBs, but it rather learns the canonicalized language model.

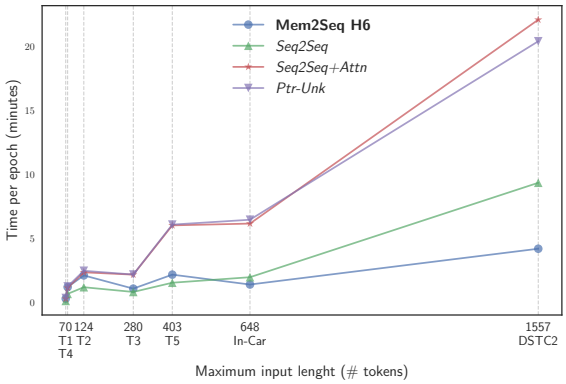

Figure 2: Training time per-epoch for different tasks (lower is better). The speed difference becomes larger as the maximal input length increases.

Time Per-Epoch: We also compare the training time [4] in Figure 2. The experiments are set with batch size 16, and we report each model with the hyper-parameter that can achieved the highest performance. One can observe that the training time is not that different for short input length (bAbI dialog tasks 1-4) and the gap becomes larger as the maximal input length increases. Mem2Seq is around 5 times faster in In-Car Assistant and DSTC2 compared to Seq2Seq with attention. This difference in training efficiency is mainly attributed to the fact that Seq2Seq models have input sequential dependencies which limit any parallelization. Moreover, it is unavoidable for Seq2Seq models to encode KBs, instead Mem2Seq only encodes with dialog history.

5 Analysis and Discussion

Memory Attention: Analyzing the attention weights has been frequently used to show the memory read-out, since it is an intuitive way to understand the model dynamics. Figure 3 shows the attention vector at the last hop for each generated token. Each column represents the P_{ptr} vector at the corresponding generation step. Our model has a sharp distribution over the memory, which im-

[4] Intel(R) Core(TM) i7-3930K CPU@3.20GHz, using a GeForce GTX 1080 Ti

Figure 3: Last hop memory attention visualization from the In-Car dataset. COR and GEN on the top are the correct response and our generated one.

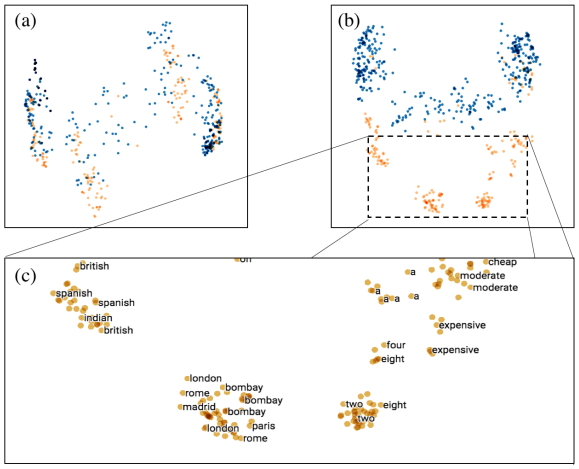

Figure 4: Principal component analysis of query vectors in hop (a) 1 and (b) 6 for bAbI dialog.

room	time	date	party	event
conference room 100	3pm	Thursday	sales team	conference
-	11am	Tuesday	-	tennis activity
-	7pm	Wednesday	-	dentist appointment
conference room 100	1pm	Tuesday	HR	meeting
-	7pm	Friday	-	doctor appointment
-	11am	Tuesday	-	lab appointment

DRIVER	When's my tennis match?
Seq2seq	Your lab appointment is on Monday at
+Att	Your lab appointment is on Tuesday at 11am
Ptr-Unk	Your tennis match is on PAD at 1pm
Mem2Seq H1	Your is very welcome
Mem2Seq H3	Your tennis is on Tuesday at 11am
Mem2Seq H6	Your tennis is Monday at 11am
GOLD	Your tennis match is Tuesday at 11am

Table 6: Example of generated responses for the In-Car Assistant on the scheduling domain.

plies that it is able to select the right token from the memory. For example, the KB information "270_altarie_walk" was retrieved at the sixth step, which is an address for "civic_center_garage". On the other hand, if the sentinel is triggered, then the generated word comes from vocabulary distribution P_{vocab}. For instance, the third generation step triggered the sentinel, and "is" is generated from the vocabulary as the word is not present in the dialog history.

Multiple Hops: Mem2Seq shows how multiple hops improve the model performance in several datasets. Task 3 in the bAbI dialog dataset serves as an example, in which the systems need to recommend restaurants to users based on restaurant ranking from highest to lowest. Users can reject the recommendation and the system has to reason over the next highest restaurant. We found out there are two common patterns between hops among different samples: 1) the first hop is usually used to score all the relevant memories and retrieve information; 2) the last hop tends to focus on a specific token and makes mistakes when the attention is not sharp. Such mistakes can be attributed to lack of hops, for some samples. For more information, we report two figures in the supplementary material.

Query Vectors: In Figure 4, the principal component analysis of Mem2Seq queries vectors is shown for different hops. Each dot is a query vector h_t during each decoding time step, and it has its corresponding generated word y_t. The blue dots are the words generated from P_{vocab}, which triggered the sentinel, and orange ones are from P_{ptr}. One can find that in (a) hop 1, there is no clear separation of two different colors but each of which tends to group together. On the other hand, the separation becomes clearer in (b) hop 6 as each color clusters into several groups such as location, cuisine, and number. Our model tends to retrieve more information in the first hop, and points into the memories in the last hop.

Examples: Table 1 and 6 show the generated responses of different models in the two test set samples from the In-Car Assistant dataset. We report examples from this dataset since their answers are more human-like and not as structured and repetitive as others. Seq2Seq generally cannot produce related information, and sometimes fail in language modeling. Instead, using attention helps with this issue, but it still rarely produces the correct entities. For example, Seq2Seq with attention generated 5 miles in Table 1 but the correct one is 4 miles. In addition, Ptr-Unk often cannot copy the correct token from the input, as shown by "PAD" in Table 1. On the other hand, Mem2Seq is able to produce the correct responses in this two examples. In particular in the navigation domain, shown in Table 1, Mem2Seq produces a different but still correct utterance. We report further examples from all the domains in the supplementary material.

Discussions: Conventional task-oriented dialog systems (Williams and Young, 2007), which are still widely used in commercial systems, require a multitude of human efforts in system designing and data collection. On the other hand, although end-to-end dialog systems are not perfect yet, they require much less human interference, especially in the dataset construction, as raw conversational text and KB information can be used directly without the need of heavy preprocessing (e.g. NER, dependency parsing). To this extent, Mem2Seq is a simple generative model that is able to incorporate KB information with promising generalization ability. We also discovered that the entity F1 score may be a more comprehensive evaluation metric than per-response accuracy or BLEU score, as humans can normally choose the right entities but have very diversified responses. Indeed, we want to highlight that humans may have a low BLEU score despite their correctness because there may not be a large n-gram overlap between the given response and the expected one. However, this does not imply that there is no correlation between BLEU score and human evaluation. In fact, unlike chat-bots and open domain dialogs where BLEU score does not correlate with human evaluation (Liu et al., 2016), in task-oriented dialogs the answers are constrained to particular entities and recurrent patterns. Thus, we believe BLEU score still can be considered as a relevant measure. In future works, several methods could be applied (e.g. Reinforcement Learning (Ranzato et al., 2016), Beam Search (Wiseman and Rush, 2016)) to improve both responses relevance and entity F1 score. However, we preferred to keep our model as simple as possible in order to show that it works well even without advanced training methods.

6 Related Works

End-to-end task-oriented dialog systems train a single model directly on text transcripts of dialogs (Wen et al., 2017; Serban et al., 2016; Williams et al., 2017; Zhao et al., 2017; Seo et al., 2017; Serban et al., 2017). Here, RNNs play an important role due to their ability to create a latent representation, avoiding the need for artificial state labels. End-to-End Memory Networks (Bordes and Weston, 2017; Sukhbaatar et al., 2015), and its variants (Liu and Perez, 2017; Wu et al., 2017, 2018) have also shown good results in such tasks. In each of these architectures, the output is produced by generating a sequence of tokens, or by selecting a set of predefined utterances.

Sequence-to-sequence (Seq2Seq) models have also been used in task-oriented dialog systems (Zhao et al., 2017). These architectures have better language modeling ability, but they do not work well in KB retrieval. Even with sophisticated attention models (Luong et al., 2015; Bahdanau et al., 2015), Seq2Seq fails to map the correct entities to the generated input. To alleviate this problem, copy augmented Seq2Seq models Eric and Manning (2017), were used. These models outperform utterance selection methods by copying relevant information directly from the KBs. Copy mechanisms has also been used in question answering tasks (Dehghani et al., 2017; He et al., 2017), neural machine translation (Gulcehre et al., 2016; Gu et al., 2016), language modeling (Merity et al., 2017), and summarization (See et al., 2017).

Less related to dialog systems, but related to our work, are the memory based decoders and the non-recurrent generative models: 1) Mem2Seq query generation phase used to access our memories can be seen as the memory controller used in Memory Augmented Neural Networks (MANN) (Graves et al., 2014, 2016). Similarly, memory encoders have been used in neural machine translation (Wang et al., 2016), and meta-learning application (Kaiser et al., 2017). However, Mem2Seq differs from these models as such: it uses multi-

hop attention in combination with copy mechanism, whereas other models use a single matrix representation. 2) non-recurrent generative models (Vaswani et al., 2017), which only rely on self-attention mechanism, are related to the multi-hop attention mechanism used in MemNN.

7 Conclusion

In this work, we present an end-to-end trainable Memory-to-Sequence model for task-oriented dialog systems. Mem2Seq combines the multi-hop attention mechanism in end-to-end memory networks with the idea of pointer networks to incorporate external information. We empirically show our model's ability to produce relevant answers using both the external KB information and the predefined vocabulary, and visualize how the multi-hop attention mechanisms help in learning correlations between memories. Mem2Seq is fast, general, and able to achieve state-of-the-art results in three different datasets.

Acknowledgments

This work is partially funded by ITS/319/16FP of Innovation Technology Commission, HKUST 16214415 & 16248016 of Hong Kong Research Grants Council, and RDC 1718050-0 of EMOS.AI.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*.

Antoine Bordes and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. *International Conference on Learning Representations*, abs/1605.07683.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS Deep Learning and Representation Learning Workshop*.

Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to attend, copy, and generate for session-based query suggestion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 1747–1756, New York, NY, USA. ACM.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49. Association for Computational Linguistics.

Mihail Eric and Christopher Manning. 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*, pages 468–473, Valencia, Spain. Association for Computational Linguistics.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *CoRR*.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149, Berlin, Germany. Association for Computational Linguistics.

Shizhu He, Cao Liu, Kang Liu, and Jun Zhao. 2017. Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 199–208, Vancouver, Canada. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. 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.

Chiori Hori, Kiyonori Ohtake, Teruhisa Misu, Hideki Kashioka, and Satoshi Nakamura. 2009. Statistical dialog management applied to wfst-based dialog systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009.*, pages 4793–4796. IEEE.

Lukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. 2017. Learning to remember rare events.

International Conference on Learning Representations.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Cheongjae Lee, Sangkeun Jung, Seokhwan Kim, and Gary Geunbae Lee. 2009. Example-based dialog modeling for practical multi-domain dialog system. *Speech Communication*, 51(5):466–484.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on speech and audio processing*, 8(1):11–23.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Fei Liu and Julien Perez. 2017. Gated end-to-end memory networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. *International Conference on Learning Representations*.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, Austin, Texas. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. *International Conference on Learning Representations*.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 583–593, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Minjoon Seo, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Query-reduction networks for question answering. *International Conference on Learning Representations*.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3784.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. 2016. Memory-enhanced decoder for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 278–286, Austin, Texas. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina Maria Rojas-Barahona, Pei hao Su, Stefan Ultes, David Vandyke, and Steve J. Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*.

Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 665–677, Vancouver, Canada. Association for Computational Linguistics.

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Genta Winata, and Pascale Fung. 2017. End-to-end recurrent entity network for entity-value independent goal-oriented dialog learning. In *Dialog System Technology Challenges Workshop, DSTC6*.

Chien-Sheng Wu, Andrea Madotto, Genta Winata, and Pascale Fung. 2018. End-to-end dynamic query memory network for entity-value independent task-oriented dialog. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

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

Tiancheng Zhao, Allen Lu, Kyusong Lee, and Maxine Eskenazi. 2017. Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 27–36. Association for Computational Linguistics.