Hierarchical-Pointer Generator Memory Network for Task Oriented Dialog

Dinesh Raghu
IBM Research, IIT Delhi
diraghu1@in.ibm.com

Nikhil Gupta
IIT Delhi
cs5140462@cse.iitd.ac.in

Mausam
IIT Delhi
mausam@cse.iitd.ac.in

Abstract

End-to-end networks trained for task-oriented dialog, such as for recommending restaurants to a user, suffer from out-of-vocabulary (OOV) problem – the entities in the Knowledge Base (KB) may not be seen by the network at training time, making it hard to use them in dialog. We propose a novel Hierarchical Pointer Generator Memory Network (HyP-MN), in which the next word may be generated from the decode vocabulary or copied from a *hierarchical memory* maintaining KB results and previous utterances. This hierarchical memory layout along with a novel KB dropout helps to alleviate the OOV problem. Evaluating over the dialog bAbI tasks, we find that HyP-MN outperforms state-of-the-art results, with considerable improvements (10% on OOV test set). HyP-MN also achieves competitive performances on various real-world datasets such as CamRest676 and In-car assistant dataset.

1 Introduction

Dialog systems that converse with the goal of accomplishing a specific task are referred to as task oriented dialog systems. Some examples include restaurant reservation (Henderson et al., 2014b), movie ticket booking (Li et al., 2017) and bus information retrieval (Raux et al., 2005). Task oriented dialogs are often grounded to a knowledge-base (KB). For example, the restaurant reservation task is grounded to a KB that contains names of restaurants, along with their details.

For effective dialog, the system should be able to gather necessary information from the user, query a KB using an *API call*, and utilize the retrieved results from the KB to generate responses. Several deep learning models have been proposed for end-to-end learning of such dialog, e.g., memory network (MN) (Bordes and Weston, 2017) and its extensions (Liu and Perez, 2017; Seo et al., 2017), and a copy augmented sequence to sequence model (Eric and Manning, 2017).

Copy augmented sequence to sequence model has the ability to copy words from the input (Vinyals et al., 2015), which could solve the OOV problem. However, the model has limited learnability as it encodes the entire dialog history and the KB tuples as a single sequence. On the other hand, memory based approaches are better at handling a large set of sequences using a multi-hop attention (Sukhbaatar et al., 2015). But their inability to copy words from the input prevents it from generating responses with unseen KB words.

To address these problems, we propose a novel Hierarchical Pointer Generator Memory Network (HyP-MN) architecture which combines multi-hop attention of a memory network with a copy mechanism. HyP-MN is enabled to copy any word from the dialog history. To accomplish this, it uses a *hierarchical memory*, which maintains two levels of addressing – at an utterance level and a word level. It also extends the bag-of-words utterance encoding of existing MN architectures by encoding an utterance as a sequence, via a bidirectional RNN. This, in conjunction with a novel KB-dropout, allows the model to effectively learn the contextual cues around each word, which is extremely useful in picking the best next word to copy.

Evaluation on dialog bAbI dataset (Bordes and Weston, 2017) shows that HyP-MN performs with a 10 point accuracy improvement when the KB has unseen words. We also perform experiments on real-world dialog datasets such as In-car Assistant (Eric et al., 2017) and CamRest676 (Wen et al., 2016), and find that HyP-MN outperforms the existing approaches on Entity F1 scores. Overall, our contributions can be summarized as:

1. We propose a novel HyP-MN architecture[1] which combines multi-hop attention of a memory network with a copy mechanism.
2. HyP-MN uses a novel hierarchical memory representation, which has the ability to represent KB tuples as a set and each utterance as a sequence of words.
3. We show state-of-the-art results on multiple datasets, including a 10% improvement on bAbI dialog dataset when the KB has unseen words .

Very recently, in parallel to our work, Madotto et al. (2018) proposed Mem2Seq, which also combines multi-hop attention with a copy mechanism. However, Mem2Seq suffers from two main problems: (1) it makes specific assumptions about how a KB fact may be structured, and only allows entities at object locations of a fact to be copied, (2) its copy mechanism is unable to utilize all the context around a word in an utterance. In contrast, HyP-MN makes no assumption about the KB structure and can copy any word from the dialog history based on its context.

2 Related Work

Dialog systems can be broadly divided into two categories: open domain (Vinyals and Le, 2015; Serban et al., 2016) and task oriented dialog systems. Open domain systems generate responses based on just the dialog history, whereas the task oriented systems generate responses based on the dialog history and a KB associated with the task.

Task oriented dialogs systems can be further divided into two: modular and end-to-end trainable dialog systems. Modular dialog systems (Wen et al., 2017; Williams et al., 2017; Williams and Young, 2007) require intermediate supervision on dialog transcripts to train each of its modules. Our work falls under end-to-end trainable dialog systems, which require just the dialog transcripts and no intermediate supervision to train. We discuss end-to-end trainable neural models along two dimensions: 1) decoding: how the response is retrieved or generated and 2) encoding: how the dialog context (dialog history and KB tuples) are represented.

Most of the existing approaches (Bordes and Weston, 2017; Liu and Perez, 2017; Seo et al., 2017; Wu et al., 2017) *retrieve* a response from a pre-defined set. These methods are generally successful when they have to provide boilerplate responses – they cannot construct new responses or use words not seen during training. To improve upon these, generative approaches are used where the next response is *generated* one word at a time (Eric and Manning, 2017; Madotto et al., 2018). These approaches also benefit the OOV problem by incorporating the ability to copy words from the input (Vinyals et al., 2015; Gu et al., 2016). This copy mechanism has also found success in summarization (Nallapati et al., 2016; See et al., 2017) and machine translation (Gulcehre et al., 2016). HyP-MN is a copy incorporated generative approach.

For encoding, existing approaches either represent the dialog context as *set of sets* (Madotto et al., 2018; Bordes and Weston, 2017; Liu and Perez, 2017) or *sequence of sequences* (Eric and Manning, 2017; Gulcehre et al., 2016). *Set of sets* represents past utterances as a bag of words making it difficult to capture context and work with words not seen during training. *Seq of seqs* enforces an order over the set of KB tuples making it harder to perform inferencing over multiple KB tuples. HyP-MN uses a *set of seqs* representation, which can both capture word contexts and perform inferencing over KB tuples.

2.1 Comparison with Mem2Seq

The closest to our model is the recently introduced Mem2Seq (Madotto et al., 2018), which also combines the multi-hop reasoning of memory networks with the generative decoding augmented with a copy mechanism. While similar in spirit to HyP-MN, there are significant differences in the two architec-

[1] We will release the code after paper acceptance

Restaurant	Cuisine	Rating	Phone
amour_bistro	french	5	ab_phone
le_cirque	french	3	lc_phone

Dialog Turn	Speaker	Utterance / KB Tuple
1	usr	hi
1	sys	hello
2	usr	can you book a table for six with french food

(a) Example KB and dialog history

m1	m2	m3	m4	m5		m15	m16	m17
amour_bistro r_phone ab_phone	5 r_rating amour_bistro	le_cirque r_phone lc_phone	3 r_rating le_cirque	t1 $u hi	…	t2 $u french	t2 $u food	sentinel

(b) Mem2Seq memory layout

m1	amour_bistro		r_phone		ab_phone	
m2	amour_bistro		r_rating		5	
m3	le_cirque		r_phone		lc_phone	
m4	le_cirque		r_rating		3	
m5	hi		$u		t1	
m6	hello		$s		t1	
m7	can	…	french	food	$u	t2

(c) HyP-MN memory layout

Figure 1: (a) an example dialog with history and KB tuples. (b) an illustration of Mem2Seq memory. (c) an illustration of HyP-MN memory

tures. We illustrate these with Figure 1, which shows an example dialog along with an associated KB in both Mem2Seq and HyP-MN memories.

First, the memory in Mem2Seq is a flat representation which operates in part at utterance level and in part at word level. In particular, the individual words of a user/machine utterance are stored in their own memory cells, whereas a KB tuple gets only one memory cell – the aggregation is done in a bag of words fashion, using words in the entire tuple. For example, the memory element *m1* contains a full KB tuple and *m15* contains only the word "french" from the second user utterance.

This implies that Mem2Seq's copy mechanism can only point to a specific KB tuple, but cannot point to a specific constituent of the tuple. This forces Mem2Seq to make a second modeling choice, of allowing only the "object" entity of a tuple to be used in the next response. These choices result in three limitations of Mem2Seq, regarding the scenarios in which Mem2Seq can be useful.

First, Mem2Seq expects each tuple to be serialized by the KB API in way that the object always comes last. This may or may not always be in control of the neural model. Second, because each utterance word has a unique memory cell. it loses the context in which the word was mentioned. As a result, the model cannot learn that "french" and other cuisines are usually followed by the word "food". This causes poor generalization where the model has to understand an OOV word from the user utterance (e.g., a new cuisine name or city name).

Finally, the model design dictates that the subject or the predicate from a KB fact can never be copied during the decode step. This adds severe limitations in practical settings. For example, the KB results in the original bAbI dataset (Bordes and Weston, 2017) came in format of *attribute(restaurant_name, restaurant_attribute)*. Since restaurant name is never an object, Mem2Seq would never be able to copy it when making recommendations. As a result, experiments with Mem2Seq necessitated a training data preprocessing, in which all rating facts had to be inverted into *rating(restaurant_rating, restaurant_name)*, so that the name could become an object and could be copied.

We believe that this adds severe limitations on the scenarios in which Mem2Seq is applicable. We perform several experiments on the original (unprocessed) datasets to highlight the loss in performance due to these modeling choices. In contrast, HyP-MN uses a hierarchical memory that can copy *any* word from the whole dialog history (see Figure 1(c)). It models each cell as a *sequence* of words, which also enables it to learn contextual cues for each word, leading to better OOV generalization.

3 The HyP-MN Architecture

The proposed Hierarchical Pointer Generator Memory Network has a multi-hop encoder and a copy augmented sequence decoder as shown in Figure 2. The network maintains all its context and KB tuples in a *hierarchical memory*.

At each time step t, the model takes the sequence of previous user utterances $\{c_1^u, \ldots, c_t^u\}$, system responses $\{c_1^s, \ldots, c_{t-1}^s\}$ and KB tuples $\{kb_1, \ldots, kb_N\}$ to generate the next system response $c_t^s = \langle y_1 y_2 \ldots y_T \rangle$ word by word.

In this section, we describe the hierarchical memory, followed by the model's architectural details. We then go on to describe the loss function, which defines the two hyper parameters used to switch con-

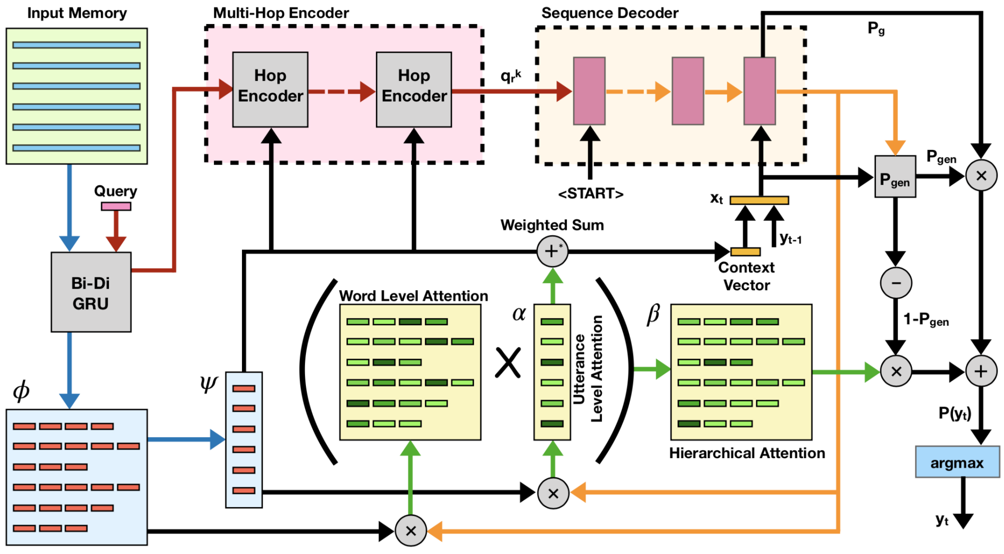

Figure 2: The encoder-decoder architecture of HyP-MN with hierarchical memory

trol between the generate and copy mode. Finally, we describe how dropout, when used in conjunction with the copy mechanism, can help make the network more immune to OOV words in the KB.

3.1 Hierarchical Memory

The memory is used by the encoder to generate a vector representation of the conversation so far, as well as by the decoder to copy words. This memory $M = \{m_i\}$ contains the dialog history $\{c^u_1, c^s_1, \dots, c^u_{t-1}, c^s_{t-1}\}$ and the KB tuples $\{kb_1, \dots, kb_N\}$ where each one is mapped to a separate memory element. In-turn, each memory element m_i is an ordered sequence of tokens $\langle w^1_i w^2_i \dots w^{|m_i|}_i \rangle$. The tokens include words in the utterance followed by a token to indicate temporal information and another to indicate the speaker information. For example, "*how may i help you #1 $s*" indicates this was the first utterance by the system.

Utterance representations are generated using a single layer bidirectional GRU. Let the outputs of the forward and backward GRUs for any word w^j_i be denoted as $\overrightarrow{h_{ij}}$ and $\overleftarrow{h_{ij}}$ respectively. The d-dimensional context dependent vector representation $\phi(.)$ of the word w^j_i, and vector representation $\psi(.)$ of the memory element m_i, are computed as follows:

$$\phi(w^j_i) = [\overrightarrow{h_{ij}}; \overleftarrow{h_{ij}}] \tag{1}$$

$$\psi(m_i) = [\overrightarrow{h_{j|m_i|}}; \overleftarrow{h_{i1}}] \tag{2}$$

3.2 The HyP-MN Encoder

The encoder used in HyP-MN is similar to the multi-hop attention encoder with layer-wise weight proposed by (Sukhbaatar et al., 2015). The encoder in (Sukhbaatar et al., 2015) uses two different embedding matrices, where as we use just one to reduce the number of parameters. The encoder takes in the last user utterance as the query $q = \psi(c^u_t)$ and computes the reduced representation q^k_r using the memory M as follows:

$$p_i = \text{softmax}(q^T \psi(m_i)) \tag{3}$$

$$o = W_r \sum\nolimits_i p_i \psi(m_i) \tag{4}$$

$$q_r = o + q \tag{5}$$

where $W_r \in \mathbb{R}^{d \times d}$. The hop step can be reiterated, by assigning the output of the previous hop as the new input query, i.e., setting $q = q_r$. The output of the encoder after K hops, q^k_r is assigned as the initial state of the HyP-MN decoder.

3.3 The HyP-MN Decoder

HyP-MN models a copy augmented sequence decoder, which generates the response one word at a time. At any time step, the decoder can either generate a word from the decode vocabulary or copy a word from the memory. To perform each decode step, the decoder computes: (1) generate distribution $P_g(y_t)$ over the decode vocabulary, (2) copy distribution $P_c(y_t)$ over the words in the memory

At any time t, the generate distribution $P_g(y_t)$ is computed using a standard sequence decoder (Sutskever et al., 2014) by attending (Luong et al., 2015) over the utterance-level representations (ULA) $alpha^t$ in the memory. The copy distribution $P_c(y_t)$ is generated by hierarchically taking a product of the word level attention (WLA) (Luong et al., 2015) and the previously computed ULA as follows:

$$\alpha_i^t = \text{softmax}(s_t W_l^u \psi(m_i)) \tag{6}$$

$$e_{ij}^t = s_t W_l^w \phi(w_i^j) \tag{7}$$

$$\beta_{ij}^t = \alpha_i^t * \frac{\exp(e_{ij}^t)}{\sum_k \exp(e_{ik}^t)} \tag{8}$$

$$P_c(y_t = w) = \sum_{ij:w_i^j=w} \beta_{ij}^t \tag{9}$$

where $W_l^w, W_l^u \in \mathbb{R}^{d\times d}$ are learnable parameters, s_t is the decoder state at time t and indicates a softmax function. The hierarchical attention helps the copy distribution to be aware of the context while generating the attention weights.

Inspired by (See et al., 2017), we combine the generate and copy distribution by using a soft gate $p_{gen} \in [0, 1]$. The final distribution is over the union of words in the decode vocabulary and the words present in the memory.

3.4 Loss

The objective is to minimize the average negative log-likelihood for all the words in the response. In addition to that we also add a cross-entropy term based on p_{gen}. The loss is given as:

$$\mathcal{L} = -\sum_{t=1}^{T} \log\Big(\mathcal{S}[p_{gen}P_g(y_t) + \rho(1-p_{gen})P_c(y_t)]\Big) + \gamma \sum_{t=1}^{T} H(p_{gen}, p_{gen}^r) \tag{10}$$

where $\mathcal{S}$ indicates a softmax function, T is the number of words in the response. This term is used to enforce copying of KB words in the response. This is achieved by using a reference distribution p_{gen}^r, which is set to 0 if word in the response is a KB word and set to 1 otherwise. The hyper-parameter ρ helps define a bias towards either copy or generate during the initial part of the training. $\rho > 1$ biases copy while $\rho < 1$ biases generation. The other hyper-parameter γ helps to define the relative importance between the two terms.

3.5 KB Dropout

As described in Section 3.1, vector representation of words $\phi(w)$ are generated using bidirectional GRUs. To help make the model robust to OOV words, it is desirable that the GRUs capture the necessary context while representing the KB words, rather than relying solely on the embedding of the exact word. We force the GRUs to learn the context, by simulating OOV words during training, i.e randomly dropping some KB words in the dialog history during each iteration. This helps the model to learn to copy KB words in the response based on the context learnt by the GRUs.

4 Experimental Setup

4.1 Datasets

We perform our experiments on four task oriented dialog datasets: bAbI dialog dataset (Bordes and Weston, 2017), DTSC2 (Henderson et al., 2014a), CamRest676 dataset (Wen et al., 2016) and In-car Assistant dataset (Eric et al., 2017).

bAbI Dialog has synthetically generated dialogs for the task of restaurant reservation. The dataset is divided into five tasks: (T1) gather user choices and generate API calls, (T2) allow the user to modify her choice and update API calls, (T3) recommend restaurants based on the options available, (T4) provide more information about the restaurant selected by the user, and (T5) all four tasks combined. The dialogs are grounded to a KB. The KB is divided into two halves. One half is used to generate train, validation and test set, while the other half is used to generate an OOV test set.

DSTC2 is a human-bot dialog dataset on restaurant reservation. Bordes and Weston (2017) formatted a dataset, which was traditionally used for evaluating modular systems, for evaluating end-to-end trainable systems.

CamRest676 is a human-human dialog dataset, also for restaurant reservation. It was collected using the Wiz-of-Oz framework. It has so far been used for evaluating modular systems. We formatted the data to evaluate end-to-end trainable systems.

In-car Assistant is also a human-human dialog dataset collected using the Wiz-of-Oz framework. Each dialog is between a driver and an in-car assistant. While other datasets have dialogs for just one task (restaurant reservation), this one has dialogs from multiple tasks (navigation, calendar and weather information).

4.2 Evaluation Metrics

We evaluate HyP-MN and other models based on their ability to generate responses. The per-response accuracy (Bordes and Weston, 2017) is the percentage of generated responses that are exactly same as the gold responses. The per-dialog accuracy is the percentage of dialogs with all the generated responses exactly same as the gold responses.

These accuracy metrics are good for evaluating dialogs with boilerplate responses such as bAbI. For other datasets, we use Entity F1 (Eric and Manning, 2017) and BLEU (Papineni et al., 2002) scores. Entity F1 is computed by taking a micro-F1 over KB entities in the entire set of gold responses. BLEU measures the overlap of n-grams between the generated response and the gold response, and has become a popular measure to compare task-oriented dialog systems. BLEU is weakly correlated with human judgments (Liu et al., 2016), but a low BLEU score across various models does indicate a high complexity of a dataset.

4.3 Baselines

We compare HyP-MN against all existing end-to-end trainable task oriented dialog systems. These include retrieval models, such as query reduction network (QRN) (Seo et al., 2017), memory network (MN) (Bordes and Weston, 2017) and gated memory network (GMN) (Liu and Perez, 2017). Retrieval models usually report improved results by including a domain-specific feature called "+match type". For a fair comparison, we keep this feature off in all retrieval models.

We also compare against generative models such as a sequence to sequence model (Seq2Seq), one augmented with Luong attention (Seq2Seq+Attn), and a copy augmented Seq2Seq+Attn (Ptr-Unk) (Gulcehre et al., 2016). For fairness across models, we do not compare against generative approaches that require additional entity type information (Eric and Manning, 2017; Eric et al., 2017), or additional intermediate labels (Wen et al., 2017).

Finally, we also compare against the recently proposed Mem2Seq (Madotto et al., 2018).[2] Results reported by Mem2Seq are not directly comparable, as they use pre-processed training data in dataset-specific ways (see Section 2.1). For direct comparisons, we re-run Mem2Seq on the original training datasets[3] (Mem2Seq*) and repeat all experiments.

4.4 Training

We train our HyP-MN network using Adam optimizer (Kingma and Ba, 2014) and apply gradient clipping with a clip-value of 40. We identify hyper-parameters by random search, evaluating on the held-out validation sets. We sample word embeddings, hidden layer, cell sizes from $\{64, 128, 256\}$ and learning rates from $\{10^{-3}, 5\times10^{-4}, 10^{-4}\}$. The hyper-parameter ρ in the loss is sampled from $\{1, 4, 8\}$ and γ in the loss is chosen between [0.5-1.5]. The KB dropout rate is sampled from $\{0.1, 0.2, 0.3\}$. The hops for multi-hop attention in the encoder is sampled from $\{1, 2, 3, 4\}$.

5 Experimental Results

Our experiments evaluate two research questions. (1) How does the performance of HyP-MN compare with existing task-oriented dialog systems (Section 5.1)? And (2) What is the incremental contribution of each of HyP-MN's components (Section 5.2)?

[2] We thank the authors for releasing a working code at `https://github.com/HLTCHKUST/Mem2Seq`

[3] Mem2Seq used the following pre-processing on the data: 1) rating facts in bAbI dialogs were inverted (subject was placed at the end) 2) an extra fact was added to the navigation tasks in In-Car Assistant with all the objects place along with the subject (subject placed at the end). We evaluated Mem2Seq* by removing these pre-processing steps.

Task	QRN	MN	GMN	Seq2Seq+Attn	Ptr-Unk	HyP-MN	Mem2Seq*	Mem2Seq
T1	99.9 (-)	99.6 (99.6)	**100 (100)**	**100 (100)**	**100 (100)**	**100 (100)**	**100 (100))**	**100 (100)**
T2	99.5 (-)	**100 (100)**	**100 (100)**	**100 (100)**	**100 (100)**	99.9 (99.8)	**100 (100)**	**100 (100)**
T3	74.8 (-)	74.9 (2.0)	74.9 (0)	74.8 (0)	85.1 (19.0)	**94.9 (63.2)**	74.9 (0)	94.7 (62.1)
T4	57.2 (-)	59.5 (3.0)	57.2 (0)	57.2 (0)	**100 (100)**	**100 (100)**	**100 (100)**	**100 (100)**
T5	**99.6 (-)**	96.1 (49.4)	96.3 (52.5)	98.4 (87.3)	99.4 (91.5)	97.7 (67)	97.7 (66.3)	97.9 (69.6)
T1-OOV	83.1 (-)	72.3 (0)	82.4 (0)	81.7 (0)	92.5 (54.7)	**100 (100)**	94.0 (62.2)	94.0 (62.2)
T2-OOV	78.9 (-)	78.9 (0)	78.9 (0)	78.9 (0)	83.2 (0)	**100 (100)**	86.5 (12.4)	86.5 (12.4)
T3-OOV	75.2 (-)	74.4 (0)	75.3 (0)	75.3 (0)	82.9 (0)	**95.6 (63.9)**	75.2 (0)	90.3 (38.7)
T4-OOV	56.9 (-)	57.6 (0)	57.0 (0)	57.0 (0)	**100 (100)**	**100 (100)**	**100 (100)**	**100 (100)**
T5-OOV	67.8 (-)	65.5 (0)	66.7 (0)	65.7 (0)	73.6 (0)	**89.3 (9.7)**	75.6 (0)	84.5 (2.3)

Table 1: Per-response and per-dialog accuracies on bAbI dialog tasks of HyP-MN and baseline models

.

5.1 Comparison against Baseline Systems

Table 1 reports the per-response and per-dialog (in parentheses) accuracies on bAbI dialog tasks. The retrieval-based models perform well on non-OOV tasks, but fail to exhibit similar performance on OOV tasks. This is expected as they are trained to retrieve from a pre-defined set of responses. Their poor non-OOV performance on tasks 3 and 4 is attributed to an error in bAbI dataset construction, due to which the KB entities present in validation and non-OOV test sets do not overlap with those in the train set. This effectively means that non-OOV and OOV test conditions are the same for tasks 3 and 4.

A simple generative model (Seq2Seq+Attn) achieves accuracies comparable to the multi-hop attention models. Enabling it with the power to copy (Ptr-Unk) shows considerable increase in performance, especially on the OOV tasks (and non-OOV tasks 3 and 4).

The good performance of sequence encoders raises a question about the value of multiple hops. HyP-MN answers this question, by obtaining vast improvements in several tasks, specifically the ones that contain OOVs. This clearly shows that both multi-hop inference and copy mechanism are essential for task oriented dialogs.

We also compare against the recently proposed Mem2Seq. As discussed, the primary comparison is against Mem2Seq*, which runs the model on unprocessed training data. We also include the numbers with pre-processing (Mem2Seq) for completeness. We find that HyP-MN outperforms both these models on all OOV tasks, except task 4, where they are at par. The difference between Mem2Seq* and Mem2Seq highlights the loss due to the assumption of copying only object words from a KB tuple. HyP-MN's further improvements over Mem2Seq are likely due to its ability to capture a word's context in an utterance better, owing to its use of bidirectional RNNs at the utterance level.

	Ent. F1	BLEU	Per Resp.	Per . Dial.
Mem2Seq	75.3	55.3	45.0	0.5
Seq2Seq	69.7	55.0	46.4	1.5
+Attn	67.1	**56.6**	46.0	1.4
Ptr-Unk	71.6	55.4	**47.3**	1.3
Mem2Seq*	**75.3**	55.3	45.0	0.5
HyP-MN	73.9	55.4	46.4	**1.7**

Table 2: Evaluations of HyP-MN and various generative models on DTSC2

	CamRest676		SMD	
	BLEU	Ent. F1	BLEU	Ent. F1
Mem2Seq	11.5	39.5	12.6	33.4
Seq2Seq	13.7	12.7	8.4	10.3
+Attn	10.2	12.5	9.3	19.9
Ptr-Unk	4.9	17.9	8.3	22.7
Mem2Seq*	11.5	39.5	**11.9**	27.3
HyP-MN	**14.06**	**43.5**	9.67	**37.4**

Table 3: Evaluations on CamRest676 and In-car Assistant datasets

Table 2 reports the results on DSTC2. HyP-MN exhibits performance comparable to other models. As the data was generated by humans interacting

	T1	T2	T3	T4	T5	T1$_{oov}$	T2$_{oov}$	T3$_{oov}$	T4$_{oov}$	T5$_{oov}$
Base Model	99.2	100	85.2	57.2	90.5	75.7	78.8	83.2	57	63.7
+ Hier. Attn.	99.9	99.9	86.9	57.2	91.2	75.6	78.8	85.3	57	65.2
+ $\gamma > 0$	99.9	99.9	88.5	57.2	91.8	75.6	78.8	86.9	57	66.1
+ $\rho > 1$	99.9	100	94.1	100	97.3	75.6	78.9	93.2	100	73.2
+ KB dropout	100	99.9	94.9	100	97.7	100	100	95.6	100	89.3

Table 4: Ablation study: incremental impact of functionalities added in HyP-MN

with bots, it contains significant noise introduced by the bot. For example, in one of the training examples, the bot responds with "You are looking for a restaurant is that right?" when the human requested for phone number of a restaurant. This inherent difficulty in the data makes it harder to model.

In Table 3, we report the numbers on two real-world datasets: CamRest676 and In-car assistant datasets. HyP-MN ranks best on both Entity F1 metric and BLEU scores on CamRest676. We discuss this further in Section 6.3. On the In-car assistant dataset, HyP-MN outperforms the baselines only on Entity F1 metric. We analyze HyP-MN's performance and observe that HyP-MN's responses often convey the necessary entity information from the KB. But, they use phrases with little lexical overlap with the gold response, reducing the BLEU scores.

5.2 Ablation Study

We assess the value of each model element, by incrementally adding functionality starting from a basic model, which is constructed using multi-hop attention to encode and copy mechanism to decode. The base model uses only WLA (as opposed to the product of WLA and ULA), is unbiased whether to generate or copy ($\rho = 1, \gamma = 0$), and trains without KB dropout. Table 4 reports the per-response accuracy scores for various configurations of HyP-MN on bAbI dialog tasks. As Task 5 is a combination of tasks 1 through 4, increase in performance on one of the first four tasks gets reflected in Task 5.

Hierarchical Attention: This choice primarily helps on Task 3. The 1.7% and 2% improvements on this task are all on the responses that recommend the best next restaurant. Performance on Task 3 is particularly important because (1) it is the only task that needs inference over the KB results. This is because the recommendation response must reason about rating orders and restaurants rejected already by the user, and recommend the highest ranked unrejected restaurant next. (2) Suggesting a restaurant is a crucial exchange in the entire task of restaurant recommendations; getting this right would be critical for the user to be satisfied with the system. We discuss the ability of hierarchical attention to better pick next restaurant in Section 6.1.

Hyper-parameters in Loss: γ and ρ play an important role in copying KB words while generating the response. By setting $\gamma > 0$ we expected the system to copy all KB words and generate the others, thus improving the numbers for tasks 3 and 4. But, to our surprise, there was only a marginal improvement on these tasks. As non-KB words in the response are higher in number compared to KB words, during initial part of the training the system tends to support generate rather than copying. This suggested that we must add a prior for preferring copying over generate. We enable this by introducing the ρ parameter. This caused the numbers for tasks 3 and 4 to shoot up, for ρ values greater than 1 (say 4 or 8).

KB word dropouts: The OOV sets of tasks 1 and 2 are still not comparable to the non-OOV sets. We find that both tasks require words to be copied from user utterances (e.g., a new cuisine), while other tasks copy only from KB tuples. The model is unable to capture the context around the words to be copied. To tackle this, we apply KB dropout and the performance improves enormously for tasks 1 and 2 (further discussion in Section 6.2).

Overall, our final model beats the existing state of the art on bAbI dataset by a significant margin. We analyze this further in the next section.

6 Discussion

6.1 Hierarchical Attention

To visualize the benefit of using a hierarchical memory, we compare attention weights over the input

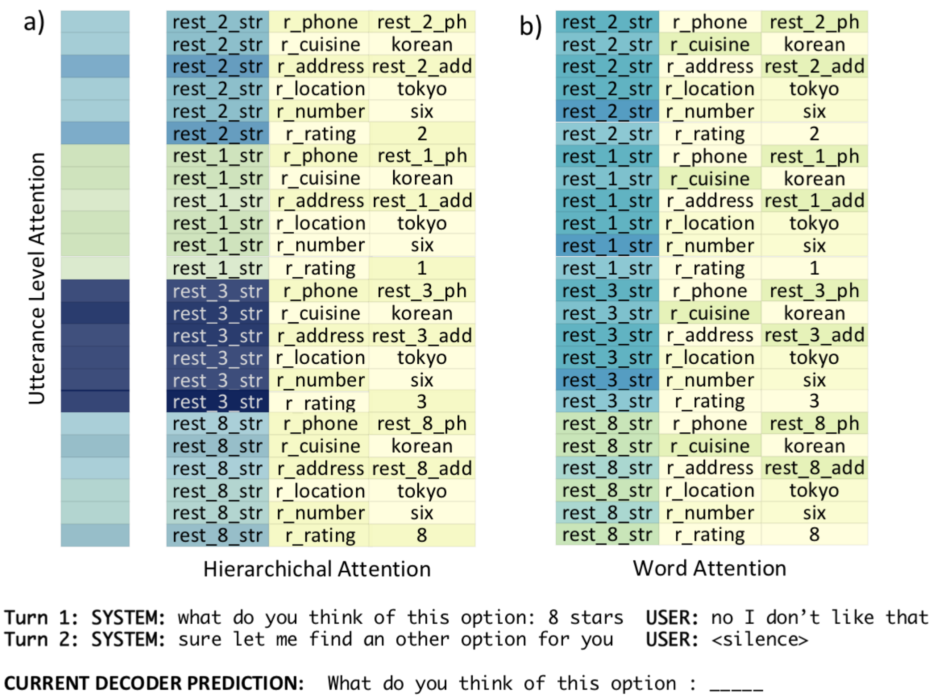

Figure 3: Visualization of attention weights on selected portions of memory in (a) HyP-MN vs (b) P-MN

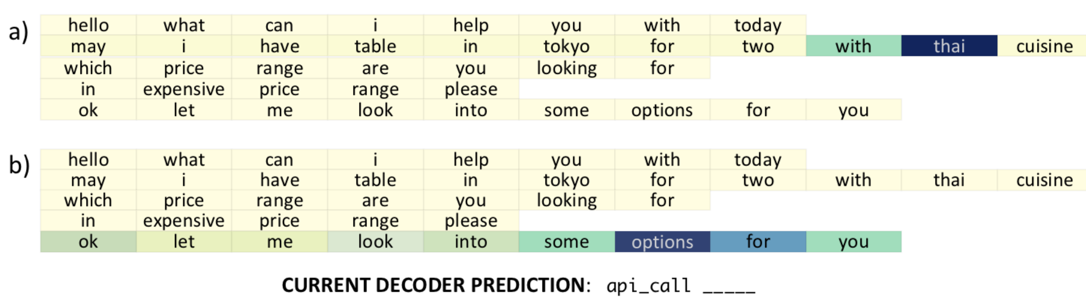

Figure 4: Visualization of attention weights on memory (subset) in HyP-MN (a) with vs. (b) without KB dropout.

context for two models: our proposed HyP-MN and a primitive variant P-MN. The only difference between P-MN and HyP-MN is the way copy distribution is computed. P-MN uses just WLA where as HyP-MNuses a product of ULA and WLA. In the example, shown in Figure 3, we are at the step of the decoder aimed at predicting the second best restaurant *3 stars*, given that the restaurant with rating *8 stars* has already been suggested and rejected. We show attention only on the KB entries for brevity.

The models share some similarities in their distribution of attention. First, the attention weights are localized over the restaurant names, indicating the preference of the system to point to a specific restaurant. This is supported by the p_{gen} values, 3.14 x 10^{-5} and 1.15 x 10^{-4} for HyP-MN and P-MN respectively, i.e., both models prefer to copy rather than generate. Moreover, entries with the same restaurant name have similar attention weights, reflecting the robustness of the distribution.

We also observe that HyP-MN is able to perform the difficult task of *sorting* the restaurant entries based on decreasing order of rating (number of stars). It gives more weight to entries with a high rating and suppresses the weights of any previously suggested restaurant, e.g., *8 stars*.

ULA provides HyP-MN with the ability to infer over multiple sets of tuples. The ability to sort the restaurants and reject a previously seen restaurant can be observed by the attention heat map of

KB (restaurant—cuisine—address—phone)	
da_vinci_pizzeria—italian—20_milton_road_chesterton—01223_351707	
usr-1 **sys-1** **usr-2**	are there any *italian* restaurants in the *north* part of town ? yes there is one , its name is *da_vinci_pizzeria* . great , can you please give me its address and phone number ?
Gold	*da_vinci_pizzeria* is located at *20_milton_road_chesterton* . their phone number is *01223_351707* . is there anything else i can help you with ?
Ptr-Unk	*da_vinci_pizzeria* is at at *2_sturton_street_city_centre* $$ their phone number is *01223_351707* .
Seq2Seq	you 're welcome .
Mem2Seq	the address is located at , and the phone number is is . is there anything else i can help you with ?
HyP-MN	phone number for *da_vinci_pizzeria* is *01223_351707* at *20_milton_road_chesterton da_vinci_pizzeria* there anything else i can help you with

Table 5: Comparing the responses generated by various models on CamRest676

ULA. WLA on the other hand can push the attention weights towards restaurants in the KB tuple. Thus using both in conjunction helps HyP-MN perform better than P-MN.

6.2 Capturing Context via KB dropout

Context plays a big hand in natural language understanding. The semantics of a word are at times ambiguous without important context clues that wrap each individual word. To successfully model our OOV tasks, the model must be able to use such clues to determine the meaning of unknown words.

In Figure 4 we compare the effect of introducing random dropouts on KB words during training of Task 1. We compare HyP-MN with and without KB dropout. In this example, the decoder has to pick the right cuisine for the API call, which itself is an OOV. We see a clear difference in attention weights on the different words of the input utterances. The dropout enhanced model is able to clearly pinpoint the never before seen cuisine from its surrounding words. The second model assigns weights to seemingly random and uninteresting words, probably owing to the fact that it relies entirely on the vector representation of a word to copy it. An unknown word like “thai” has no learnt representation for the model to recognize it as a cuisine.

6.3 Qualitative Analysis of Real Datasets

We qualitatively analyze the dialogs generated for real-world datasets. In Table 5 we compare the output responses from four different models. The important KB entities in each response have been italicized for emphasis.

The dialog systems are expected provide the user with requested information – address and phone number of *da_vinci_pizzeria*. Ptr-Unk is able to point to the requested phone number but generates an incorrect address. Seq2Seq generates a response irrelevant to the dialog context. Mem2Seq has learnt to generate a very readable response, similar to the ground truth but fails to pick the necessary entities from the KB. Finally, HyP-MN is able to copy necessary entities – address and phone number of *da_vinci_pizzeria* (high Entity F1 score) while generating the response, but doesnot generate the most readable response. However, it manages to generate the most informative and useful response to the user.

7 Conclusions

We propose a Hierarchical-Pointer Generator Memory Network (HyP-MN) for training task-oriented dialog systems in an end to end fashion. HyP-MN combines several innovations over a basic memory network architecture, including (1) a hierarchical memory that stores a dialog context (utterances, KB tuples) in a bag of sequences representation, (2) a copy-augmented generative decoder that uses hierarchical attention over context to copy words in its response, (3) loss-augmentation and hyper-parameters to encourage the model to copy when it can also generate the same word, and (4) a KB-specific dropout that randomly drops out KB words appearing in a dialog, thereby learning more effective contextual cues. A combination of these features enables the model to be extremely effective in making use of out-of-vocabulary entities in its responses, whether they were mentioned in KB tuples or user utterances.

HyP-MN achieves the state of the art results on bAbI dialog dataset, outperforming existing models by 10 points or more on the OOV conditions. Experiments on real world datasets show that HyP-MN employs important entities in the dialog much better than other systems. We will release our code for further research.

References

Antoine Bordes and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *International Conference on Learning Representations*.

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. Association for Computational Linguistics.

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, Saarbrücken, Germany, August 15-17, 2017*, pages 37–49.

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. 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. Association for Computational Linguistics.

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

Matthew Henderson, Blaise Thomson, and Steve Young. 2014b. Word-based dialog state tracking with re- current neural networks. In *In Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299.

Diederik P Kingma and Jimmy Lei Ba. 2014. Adam: Amethod for stochastic optimization. In *Proc. 3rd Int. Conf. Learn. Representations*.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, pages 733–743. Asian Federation of Natural Language Processing.

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*, volume 1, pages 1–10.

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.

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. Association for Computational Linguistics.

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

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290. 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 the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Antoine Raux, Brian Langner, Dan Bohus, Alan W. Black, and Maxine Eskénazi. 2005. Let's go public! taking a spoken dialog system to the real world. In *INTERSPEECH*.

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. Association for Computational Linguistics.

Minjoon Seo, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Query-reduction networks for question answering. In *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.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *Proceedings of the International Conference on Machine Learning, Deep Learning Workshop.*

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Lina M. Rojas Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2016. Conditional generation and snapshot learning in neural dialogue systems. In *EMNLP*, pages 2153–2162, Austin, Texas, November. ACL.

TH Wen, D Vandyke, N Mrkšíc, M Gašíc, LM Rojas-Barahona, PH Su, S Ultes, and S Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017-Proceedings of Conference*, volume 1, pages 438–449.

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. volume 21, pages 393–422. Elsevier.

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)*, volume 1, pages 665–677.

Chien-Sheng Wu, Andrea Madotto, G 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*.