Disentangling Language and Knowledge in Task-Oriented Dialogs

Dinesh Raghu[*1 2], Nikhil Gupta[1] and Mausam[1]

[1] IIT Delhi, New Delhi, India
[2] IBM Research, New Delhi, India
diraghu1@in.ibm.com, nikhilgupta1997@gmail.com, mausam@cse.iitd.ac.in

Abstract

The Knowledge Base (KB) used for real-world applications, such as booking a movie or restaurant reservation, keeps changing over time. End-to-end neural networks trained for these task-oriented dialogs are expected to be immune to any changes in the KB. However, existing approaches breakdown when asked to handle such changes. We propose an encoder-decoder architecture (BOSSNET) with a novel Bag-of-Sequences (BOSS) memory, which facilitates the disentangled learning of the response's language model and its knowledge incorporation. Consequently, the KB can be modified with new knowledge without a drop in interpretability. We find that BOSSNET outperforms state-of-the-art models, with considerable improvements ($>$10%) on bAbI OOV test sets and other human-human datasets. We also systematically modify existing datasets to measure disentanglement and show BOSSNET to be robust to KB modifications.

1 Introduction

Task-oriented dialog agents converse with a user with the goal of accomplishing a specific task and often interact with a knowledge-base (KB). For example, a restaurant reservation agent (Henderson et al., 2014) will be grounded to a KB that contains the names of restaurants, and their details.

In real-world applications, the KB information could change over time. For example, (1) a KB associated with a movie ticket booking system gets updated every week based on new film releases, and (2) a restaurant reservation agent, trained with the knowledge of eateries in one city, may be deployed in other cities with an entirely different range of establishments. In such situations, the system should have the ability to conform to new-found knowledge unseen during its training. Ideally, the training algorithm must learn to disentangle the language
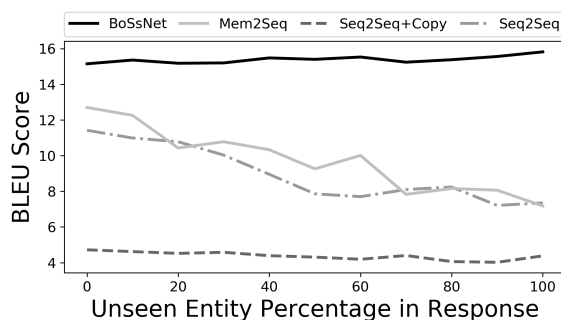

Figure 1: Performance of various task-oriented dialog systems on the CamRest dataset as the percentage of unseen information in the KB changes.

model from the knowledge interface model. This separation will enable the system to generalize to KB modifications, without a loss in performance.

Moreover, for achieving good progress towards the user's task, the agent must also retain the ability to draw inferences based on past utterances and the KB. Notably, we find that existing approaches either achieve this disentanglement or effective progress towards the task, but not both.

For instance, Mem2Seq (Madotto et al., 2018) exhibits satisfactory performance when tested on the training KB. It represents the dialog history and the KB knowledge as a *bag of words* in a flat memory arrangement. This enables Mem2Seq to revisit each word several times, as needed, obtaining good performance. But at the same time, flat memory prevents it from capturing any surrounding context – this deteriorates its performance rapidly when the amount of new unseen information in the KB increases, as shown in Figure 1. On the other hand, the performance of copy augmented sequence-to-sequence network (Seq2Seq+Copy) (Eric and Manning, 2017), is robust to changes in the KB, but fails to achieve acceptable task-oriented performance. It captures context by representing the entire dialog history as one continuous *sequence*. However, it can be difficult for a sequence encoder to reason

over long dialogs found in real-world datasets and its ability to learn the task gets hampered.

We propose BOSSNET, a novel network that effectively disentangles the language and knowledge models, and also achieves state-of-the-art performance on three existing datasets.

To achieve this, BOSSNET makes two design choices. First, it encodes the conversational input as a *bag of sequences* (BOSS) memory, in which the input representation is built at two levels of abstraction. The *higher level* flat memory encodes the KB tuples and utterances to facilitate effective inferencing over them. The *lower level* encoding of each individual utterance and tuple is constructed via a sequence encoder (Bi-GRU). This enables the model to maintain the sequential context surrounding each token, aiding in better interpretation of unseen tokens at test time. Second, we augment the standard cross-entropy loss used in dialog systems with an additional loss term to encourage the model to only copy KB tokens in a response, instead of generating them via the language model. This combination of sequence encoding and additional loss (along with dropout) helps in effective disentangling between language and knowledge.

We perform evaluations over three datasets – bAbI (Bordes and Weston, 2017), CamRest (Wen et al., 2016), and Stanford Multi-Domain Dataset (Eric et al., 2017). Of these, the last two are real-world datasets. We find that BOSSNET is competitive or significantly better on standard metrics in all datasets as compared to state-of-the-art baselines. We also introduce a *knowledge adaptability* (KA) evaluation, in which we systematically increase the percentage of previously unseen entities in the KB. We find that BOSSNET is highly robust across all percentage levels. Finally, we also report a human-based evaluation and find that BOSSNET responses are frequently rated higher than other baselines.

Overall, our contributions are:

1. We propose BOSSNET, a novel architecture to disentangle the language model from knowledge incorporation in task-oriented dialogs.

2. We introduce a *knowledge adaptability* evaluation to measure the ability of dialog systems to scale performance to unseen KB entities.

3. Our experiments show that BOSSNET is competitive or significantly better, measured via standard metrics, than the existing baselines on three datasets.
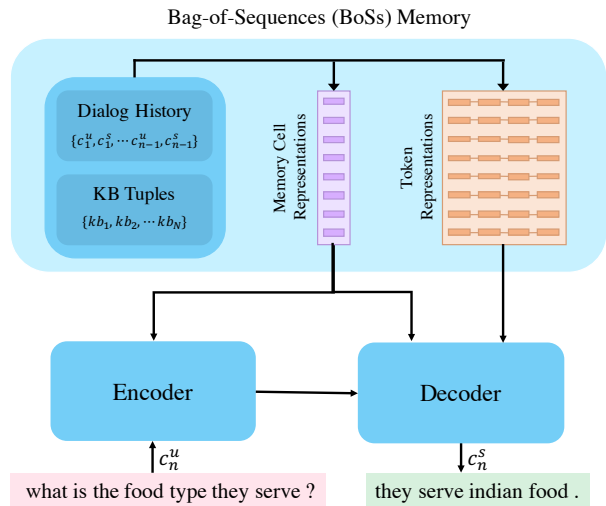

Figure 2: The dialog history and KB tuples stored in the memory have memory cell representations and token representations. The encoder understands the last user utterance using only the memory cell representations. The decoder generates the next response using both representations.

We release our code and *knowledge adaptability* (KA) test sets for further use by the research community.[1]

2 The BOSSNET Architecture

The proposed Bag-of-Sequences Memory Network has an encoder-decoder architecture that takes as input (1) dialog history, which includes a sequence of previous user utterances $\{c_1^u, \ldots, c_n^u\}$ and system responses $\{c_1^s, \ldots, c_{n-1}^s\}$, and (2) KB tuples $\{kb_1, \ldots, kb_N\}$. The network then generates the next system response $c_n^s = \langle y_1 y_2 \ldots y_T \rangle$ word-by-word. The simplified architecture of BOSSNET is shown in Figure 2.

In this section, we first describe the BOSS memory which contains the dialog history and KB tuples, followed by how the memory is consumed by the encoder and the decoder. We finally define the loss function, which, along with dropout, enables disentangled learning of language and knowledge.

2.1 Bag-of-Sequences Memory

The memory M contains the dialog history $\{c_1^u, c_1^s, \ldots, c_{n-1}^u, c_{n-1}^s\}$ and the KB tuples $\{kb_1, \ldots, kb_N\}$. Each utterance in the dialog history and each KB tuple is placed in a memory cell. As utterances and tuples are inherently a sequence, we represent each memory cell m_i as

---
[1] https://github.com/dair-iitd/BossNet

an ordered sequence of tokens $\langle w_i^1 w_i^2 \ldots w_i^{|m_i|} \rangle$. For an utterance, the word tokens are followed by a temporal indicator and a speaker indicator $\{\$u, \$s\}$. For example, $\{$good, morning, #1, $s\}$ indicates this was the first utterance by the system. For a KB tuple, the tokens are sequenced as $\{subject, predicate, object\}$ followed by temporal indicator and a kb indicator (db).

Token representation is generated using a bidirectional GRU. Let the outputs of the forward and backward GRUs for the token w_i^j be denoted as $\overrightarrow{h_i^j}$ and $\overleftarrow{h_i^j}$ respectively. Then the token representation $\phi(w_i^j)$ is given by Eq. 1. Memory cell representation $\psi(m_i)$ is computed by concatenating the forward GRU output of its last token and the backward GRU output of its first token as in Eq. 2.

$$\phi(w_i^j) = [\overrightarrow{h_i^j}; \overleftarrow{h_i^j}] \tag{1}$$

$$\psi(m_i) = [\overrightarrow{h_i^{|m_i|}}; \overleftarrow{h_i^1}] \tag{2}$$

2.2 The BOSSNET Encoder

The encoder used in BOSSNET is similar to the multi-hop attention encoder with layer-wise weights proposed by Sukhbaatar et al. (2015). The encoder in Sukhbaatar et al. (2015) uses two different embedding matrices, whereas we use just one to reduce the number of parameters. The encoder considers the last user utterance as the query $q = \psi(c_n^u)$ and computes the reduced representation q_r using the memory M as follows:

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

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

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

where $W_r, W_o \in \mathbb{R}^{d \times d}$ are learnable parameters. The hop step can be re-iterated, 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_r^k, is assigned as the initial state of the BOSSNET decoder.

2.3 The BOSSNET Decoder

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

The generate distribution is computed using a standard sequence decoder (Sutskever et al., 2014) by attending (Luong et al., 2015) over the memory cell representations ψ. The copy distribution is generated by using a *two-level attention*. Given the decoder state s_t, it first computes attention α_t over the memory cells. Then it computes attention over the tokens in each memory cell m_i. Finally it multiplies both these attentions to compute $P_c(y_t)$ as follows:

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

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

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

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

The copy and generate distributions are combined using a soft gate $g_s^t \in [0, 1]$ as in See et al. (2017). g_s^t is a function of the decoder state at time t and the word decoded in the previous time step.

2.4 Loss

The decoder is trained using cross-entropy loss. The loss per response is defined as:

$$\mathcal{L}_{ce} = -\sum_{t=1}^{T} \log\left(g_s^t P_g(y_t) + (1 - g_s^t) P_c(y_t) \right) \tag{10}$$

where T is the number of words in the sequence to be generated and y_t is the word to be generated at time step t. The decision to generate or copy is learnt implicitly by the network. However, to attain perfect disentanglement, the KB words should be copied, while the language should be generated. In other words, any word in the response that is present in the BOSS KB memory should have a low g_s. To obtain this behavior, we define a disentangle label D_l for each word in the response. This label is set to 1 if the word is present in the BOSS KB memory and 0 otherwise. We define a disentangle loss as follows:

$$\mathcal{L}_d = -\sum_{t=1}^{T} g_s^t \log D_l^t + (1 - g_s^t) \log(1 - D_l^t) \tag{11}$$

We randomly drop some words with disentangle label set to 1. This *Disentangle Label Dropout (DLD)* works in tandem with the disentangle loss and BOSS memory – it encourages the model to

copy KB words whenever possible, based on their surrounding words. The overall loss is given as:

$$\mathcal{L} = \mathcal{L}_{ce} + \gamma \mathcal{L}_d \qquad (12)$$

The relative weight of $\mathcal{L}_d$ in the overall loss is controlled using a hyper-parameter (γ). The dropout rate is also a hyper-parameter.

3 Experimental Setup

We perform experiments on three task-oriented dialog datasets: bAbI Dialog (Bordes and Weston, 2017), CamRest (Wen et al., 2016), and Stanford Multi-Domain Dataset (Eric et al., 2017).

bAbI Dialog consists of synthetically generated dialogs with the goal of restaurant reservation. The dataset consists of five different tasks, all grounded to a KB. This KB is split into two mutually exclusive halves. One half is used to generate the train, validation, and test sets, while the other half is used to create a second test set called the OOV test set.

CamRest is a human-human dialog dataset, collected using the Wiz-of-Oz framework, also aimed at restaurant reservation. It is typically used to evaluate traditional slot filling systems. In order to make it suitable for end-to-end learning, we stripped the handcrafted state representations and annotations in each dialog, and divided the 676 available dialogs into train, validation, and test sets (406, 135, and 135 dialogs, respectively).

Stanford Multi-Domain Dataset (SMD) is another human-human dialog dataset collected using the Wiz-of-Oz framework. Each conversation is between a driver and an in-car assistant. The other datasets consist of dialogs from just one domain (restaurant reservation), whereas SMD consists of dialogs from multiple domains (calendar scheduling, weather information retrieval, and navigation).

3.1 Knowledge Adaptability (KA) Test Sets

Each bAbI dialog task has an additional OOV test set, which helps to evaluate a model's robustness to change in information in the KB. A model that perfectly disentangles language and knowledge should have no drop in accuracy on the OOV test set when compared to the non-OOV test set. To measure the degree of disentanglement in a model, we generated 10 additional test sets for each real-world corpus by varying the percentage (in multiples of 10) of unseen entities in the KB. We systematically picked random KB entities and replaced all their occurrences in the dialog with new entity names. We will refer to these generated dialogs as the *Knowledge Adaptability* (KA) test sets.

3.2 Baselines

We compare BOSSNET against several existing end-to-end task-oriented dialog systems. These include retrieval models, such as the query reduction network (QRN) (Seo et al., 2017), memory network (MN) (Bordes and Weston, 2017), and gated memory network (GMN) (Liu and Perez, 2017). We also compare against generative models such as a sequence-to-sequence model (Seq2Seq), a copy augmented Seq2Seq (Seq2Seq+Copy) (Gulcehre et al., 2016), and Mem2Seq (Madotto et al., 2018).[2] For fairness across models, we do not compare against key-value retrieval networks (Eric et al., 2017) as they simplify the dataset by canonicalizing all KB words in dialogs.

We noticed that the reported results in the Mem2Seq paper are not directly comparable, as they pre-processed[3] training data in SMD and bAbI datasets. For fair comparisons, we re-run Mem2Seq on the original training datasets. For completeness we mention their reported results (with pre-processing) as Mem2Seq*.

3.3 Evaluation Metrics

We evaluate BOSSNET and other models based on their ability to generate valid responses. The per-response accuracy (Bordes and Weston, 2017) is the percentage of generated responses that exactly match their respective gold response. The per-dialog accuracy is the percentage of dialogs with all correctly generated responses. These accuracy metrics are a good measure for evaluating datasets with boilerplate responses such as bAbI.

To quantify performance on other datasets, we use BLEU (Papineni et al., 2002) and Entity F1 (Eric and Manning, 2017) scores. BLEU measures the overlap of n-grams between the generated response and its gold response and has become a popular measure to compare task-oriented dialog systems. Entity F1 is computed by micro-F1 over KB entities in the entire set of gold responses.

---

[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) The subject (restaurant name) and object (rating) positions of the rating KB tuples in bAbI dialogs are flipped 2) An extra fact was added to the navigation tasks in SMD which included all the properties (distance, address, etc.) combined together as the subject and *poi* as the object. See Appendix.

3.4 Human Evaluation

We use two human evaluation experiments to compare (1) the *usefulness* of a generated response with respect to solving the given task, and (2) the *grammatical correctness* and *fluency* of the responses on a 0–3 scale. We obtain human annotations by creating Human Intelligence Tasks (HITs) on Amazon Mechanical Turk (AMT). For each test condition (percentage of unseen entities), we sampled 50 dialogs from Camrest and SMD each, and two AMT workers labeled each system response for both experiments, resulting in 200 labels per condition per dataset per system. We evaluate four systems in this study, leading to a total of 1600 labels per condition. The detailed setup is given in the Appendix.

3.5 Training

We train BOSSNET using an Adam optimizer (Kingma and Ba, 2014) and apply gradient clipping with a clip-value of 40. We identify hyperparameters based on the evaluation of the held-out validation sets. We sample word embedding, hidden layer, and cell sizes from $\{64, 128, 256\}$ and learning rates from $\{10^{-3}, 5 \times 10^{-4}, 10^{-4}\}$. The hyper-parameter γ in the loss function is chosen between [0-1.5]. The Disentangle Label Dropout rate is sampled from $\{0.1, 0.2\}$. The number of hops for multi-hop attention in the encoder is sampled from $\{1, 3, 6\}$. The best hyper-parameter setting for each dataset is reported in the Appendix.

4 Experimental Results

Our experiments evaluate three research questions.
1. *Performance Study*: How well is BOSSNET able to perform the tasks of our three datasets as compared to the baseline models?
2. *Disentanglement Study*: How robust are the models in generalizing on the KA test sets?
3. *Ablation Study*: What is the performance gain from each novel feature in BOSSNET?

4.1 Performance Study

Table 1 reports the per-response and per-dialog (in parentheses) accuracies on the bAbI dialog tasks. The multi-hop retrieval-based models such as QRN, MN and GMN perform well on the non-OOV test sets for tasks 1, 2, and 5, but fail to exhibit similar performance on the corresponding OOV test sets. This result is expected as these models 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 the bAbI dataset construction, due to which, the non-OOV and OOV test conditions are the same for these tasks (see Appendix).

A simple generative model (Seq2Seq) achieves accuracies comparable to the multi-hop retrieval models. Enabling it with the ability to copy from the context (Seq2Seq+Copy) shows a considerable increase in performance, especially on the OOV test sets (and non-OOV tests for tasks 3 and 4).

The strong performance of simple sequence encoders when compared with multi-hop encoders (in retrieval models) raises a question about the value of multi-hop inference. Mem2Seq answers this question, by obtaining improvements in several tasks, specifically on their OOV test sets. This clearly shows that multi-hop inference and the copy mechanism are essentials for task-oriented dialogs.

Despite gains from the Mem2Seq model, the performance difference between the non-OOV and OOV test sets remains large. BOSSNET succeeds to bridge this gap with its ability to better interpret unseen words, using their surrounding context. It obtains significant improvements on average of about 34% per-dialog accuracy and 10% per-response accuracy for the bAbI OOV test sets.

In Table 2, we report results on the real-world datasets. BOSSNET greatly outperforms other models in both Entity F1 metric and BLEU scores on CamRest. On SMD, BOSSNET achieves the best only in Entity F1. On further analysis of the generated responses we observe that BOSSNET responses often convey the necessary entity information from the KB. However, they consist of meaningful phrases with little lexical overlap with the gold response, reducing the BLEU scores. We investigate this further in our human evaluation.

Human Evaluation: We summarize the human evaluation results for real-world datasets in Table 3. BOSSNET shows the best performance on Camrest, and is judged useful 77 times out of 100. Also, it has the highest average grammatical correctness score of 2.28 (very close to Seq2Seq and Mem2Seq). BOSSNET performs on par with Mem2Seq and Seq2Seq in its ability to relay appropriate information to solve SMD dialog tasks, and has a slightly higher grammaticality score.

4.2 Disentanglement Study

We use our generated knowledge adaptability (KA) test sets to measure the robustness of BOSSNET

Task	Retrieval Models			Generative Models				
	QRN	MN	GMN	Mem2Seq*	Seq2Seq	Seq2Seq+Copy	Mem2Seq	BOSSNET
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)	100 (100)	100 (100)	100 (100)
T3	74.8 (-)	74.9 (2.0)	74.9 (0)	94.7 (62.1)	74.8 (0)	85.1 (19.0)	74.9 (0)	**95.2 (63.8)**
T4	57.2 (-)	59.5 (3.0)	57.2 (0)	100 (100)	57.2 (0)	100 (100)	100 (100)	100 (100)
T5	**99.6 (-)**	96.1 (49.4)	96.3 (52.5)	97.9 (69.6)	97.2 (64.4)	96 (49.1)	97.7 (66.3)	97.3 (65.6)
T1-OOV	83.1 (-)	72.3 (0)	82.4 (0)	94.0 (62.2)	81.7 (0)	92.5 (54.7)	94.0 (62.2)	**100 (100)**
T2-OOV	78.9 (-)	78.9 (0)	78.9 (0)	86.5 (12.4)	78.9 (0)	83.2 (0)	86.5 (12.4)	**100 (100)**
T3-OOV	75.2 (-)	74.4 (0)	75.3 (0)	90.3 (38.7)	75.3 (0)	82.9 (0)	75.2 (0)	**95.7 (66.6)**
T4-OOV	56.9 (-)	57.6 (0)	57.0 (0)	100 (100)	57.0 (0)	100 (100)	100 (100)	100 (100)
T5-OOV	67.8 (-)	65.5 (0)	66.7 (0)	84.5 (2.3)	67.4 (0)	73.6 (0)	75.6 (0)	**91.7 (18.5)**

Table 1: Per-response and per-dialog accuracies (in brackets) on bAbI dialog tasks of BOSSNET and baselines
.

	CamRest		SMD	
	BLEU	Ent. F1	BLEU	Ent. F1
Mem2Seq*	12.7	39	12.6	33.4
Seq2Seq	11.4	40.6	8.7	34.9
Seq2Seq+Copy	4.7	32.2	3.23	16.9
Mem2Seq	12.7	39	**10.3**	31.8
BOSSNET	**15.2**	**43.1**	8.3	**35.9**

Table 2: Performance of BOSSNET and baselines on
the CamRest and SMD datasets

	CamRest		SMD	
	Info	Grammar	Info	Grammar
Seq2Seq	46	2.24	35	2.38
Seq2Seq+Copy	27	1.1	21	1.04
Mem2Seq	51	2.2	**38**	2.0
BOSSNET	**77**	**2.28**	36	**2.5**

Table 3: AMT Evaluations on CamRest and SMD

KB (restaurant—address)
r_bangkok_overpriced_thai_8—r_bangkok_overpriced_thai_8_addr
r_bangkok_overpriced_thai_7—r_bangkok_overpriced_thai_7_addr
r_bangkok_overpriced_thai_4—r_bangkok_overpriced_thai_4_addr
r_bangkok_overpriced_thai_2—r_bangkok_overpriced_thai_2_addr

usr-1	may i have a table in an *overpriced* price range for *nine* people with *thai* food in *bangkok* ?
sys-1	what do you think of : *r_bangkok_overpriced_thai_8* ?
usr-2	can you provide the address ?
Gold	here it is *r_bangkok_overpriced_thai_8_addr*
Seq2Seq+Copy	here it is *r_bangkok_overpriced_thai_4_addr*
Seq2Seq	here it is *r_london_moderate_spanish_6_addr*
Mem2Seq	here it is *r_bangkok_overpriced_thai_4_addr*
BOSSNET	here it is *r_bangkok_overpriced_thai_4_addr*

Table 4: Example from bAbI Task 5 KA test set with
100% OOV entities. Identifying the address of an un-
seen restaurant is challenging for all models.

and the other baselines to changes in the KB.
We perform this experiment on 4 different tasks,
namely bAbI tasks 1 and 5, CamRest, and SMD.

Figures 3 and 4 show the per-response accura-
cies of the two bAbI dialog tasks plotted against
the percentage of unseen entities in KA sets. From
Figure 3 we observe that BOSSNET remains im-
mune to any variablity in the KB content, whereas
the performance of Mem2Seq and Seq2Seq models
drops drastically due to their inability to capture
semantic representations of the injected KB enti-
ties. We see a similar trend in Figure 4, but here
all the models show a drop in performance, with
BOSSNET appearing the most steady. We explain
this trend using the example dialog in Table 4. In
the current dialog context, the system is required to
provide the address of the selected restaurant, but
since more than one restaurant in the KB is unseen,
it becomes ambiguous for the network to identify

the correct restaurant and infer its address. In the
end, the system is forced to pick a random address
– the probability of which being correct decreases
as more restaurants become unseen.

The performance on the CamRest KA test sets
is illustrated in Figures 1 and 5. BOSSNET has
the best performance with even a slight increase in
both BLEU and Entity F1 metrics as more OOV
content is injected in the dialog, probably because
it is clear that it needs to copy when processing
unseen entities. Seq2Seq+Copy is unable to per-
form well in CamRest as the length of the input
(dialog history + KB tuples) is long and the size
of the training set is also small. We believe that
Seq2Seq+Copy works best in an environment with
an abundance of short dialog training data (e.g.,
bAbI task 1 in Figure 3).

SMD consists of dialogs with a large KB and a
highly varying response pattern. This makes it very
difficult to learn the language model – reflected in
the low BLEU scores for all the systems. BOSS-
NET still provides the best F1 entity score due to

	CamRest		SMD	
	Info	Grammar	Info	Grammar
Seq2Seq	26	2.28	22	**2.44**
Seq2Seq+Copy	22	1.22	16	1.04
Mem2Seq	35	2.06	26	1.9
BOSSNET	**80**	**2.44**	**51**	2.28

Table 5: AMT Evaluations on CamRest and SMD (50% unseen) KA datasets

its ability to inference efficiently on the large KB (Figure 6). Mem2Seq shows the best BLEU score performance on the original test set, but its performance drop of 42.5%, from 10.3 at 0% unseen to 5.93 at 100% unseen, is a lot heavier than that of BOSSNET which only drops 7.6% – 8.27 at 0% unseen to 7.64 at 100% unseen.

Human Evaluation: We summarize the human evaluation results for real-world datasets on the 50% unseen KA test set in Table 5. BOSSNET again outperforms the baselines and is labeled *successful* twice more often than the next best model on both Camrest and SMD. Seq2Seq appears to produce better sentence structures on the SMD dataset, primarily because it does not attempt to learn inference on the KB, allowing it to solely focus on learning the language model better.

4.3 Ablation Study

We assess the value of each model element, by removing it from BOSSNET. Table 6 reports the per-response accuracy scores for various configurations of BOSSNET on bAbI dialog tasks. It also reports the BLEU and entity F1 metric of various configurations on CamRest.

Without BoSs Memory: This configuration uses the Bag-of-Bags (BoB) Memory rather than BOSS memory. The BoB memory is a simplified representation, similar to the one in the original Memory Networks. Here the token representation is the vector embedding of the token with no influence from the surrounding words and the memory cell representation is the sum of all its token embeddings. As a result, each word w representation is influenced equally by all words in a memory cell, irrespective of its distance from w. This makes capturing context in the immediate neighbourhood harder. Inability to capture the correct context prevents the configuartion from generalizing to OOV test sets.

Without Disentangled Loss: Disentangled Loss ($\mathcal{L}_d$) plays an important role in enforcing that KB words be copied and other language be generated.

By removing this loss component, it achieves better BLEU score in CamRest, but with a drop in Entity F1. Without the disentangled loss, the model sometimes learns to generate KB words. This severely affects OOV performance. As described earlier, an error in bAbI dataset construction tasks 3 and 4 effectively injects the validation set with a lot of OOVs. This anomaly in conjunction with the dropout (DLD), helps the configuration in achieving an acceptable performance for those tasks.

Without Disentangled Label Dropout: BOSSNET learns to generate language and copy KB words. Without DLD, the model learns to memorize words to be copied rather than learning the context under which a word should be copied. Hence, the performance on OOV test sets is much inferior compared to the non-OOV setting.

Overall, we notice that combining all three model elements is necessary in obtaining the best performance across all tasks.

4.4 Qualitative Evaluation

We qualitatively compare the performance of BOSSNET with other baselines using examples.

Table 7, demonstrates the ability of BOSSNET to copy entities (restaurant name and address) in its response. The other baselines either generate unwanted or irrelevant entities in their response, or fail to copy altogether. BOSSNET also best captures the language model effectively with a slight paraphrasing of the gold response.

Table 8 contains only unseen entities. This example highlights the shortcomings of the Seq2Seq model as it ends up predicting a restaurant encountered during training. Mem2Seq copies a restaurant name without learning to sort the restaurants based on rating. BOSSNET, with its efficient memory addressing, is seen to be able to solve both issues.

5 Related Work

Compared to the traditional slot-filling based dialog (Williams and Young, 2007; Wen et al., 2017; Williams et al., 2017), end-to-end training methods (e.g., (Bordes and Weston, 2017), this work) do not require handcrafted state representations and their corresponding annotations in each dialog. Thus, they can easily be adapted to a new domain. We discuss end-to-end approaches along two verticals: 1) decoder: whether the response is retrieved or generated and 2) encoder: how the dialog history and KB tuples are encoded.
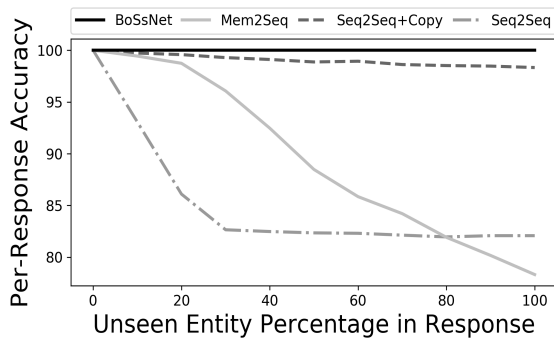
Figure 3: bAbI Task 1: Per-response accuracy comparison on KA sets
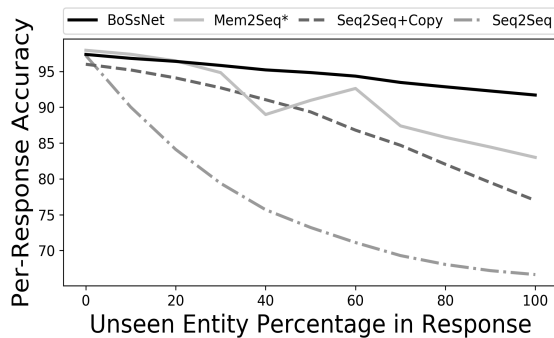

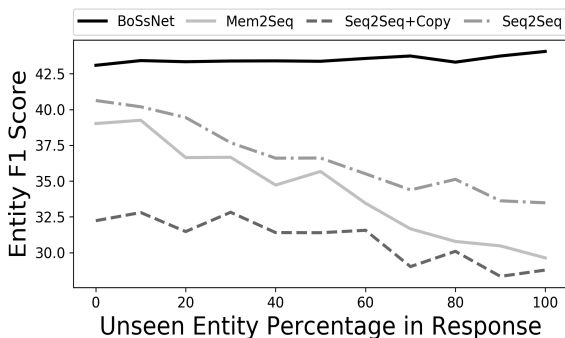Figure 4: bAbI Task 5: Per-response accuracy comparison on KA sets

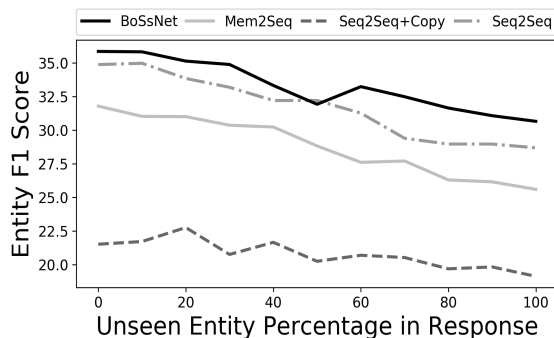Figure 5: CamRest: Entity F1 comparison on KA sets

Figure 6: SMD: Entity F1 comparison on KA sets

	bAbI Dialog Tasks					bAbI Dialog Tasks (OOV)					CamRest	
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	BLEU	Ent. F1
BOSSNET w/o BOSS Memory	100	100	74.9	57.2	95.6	93.5	78.9	74.9	57	81.4	10.13	29
BOSSNET w/o $\mathcal{L}_d$	100	100	91.7	100	94.3	83.2	78.9	92.7	100	66.7	15.5	40.1
BOSSNET w/o DLD	100	100	93.4	100	95.3	79.2	84.6	90.7	100	78.1	12.4	40.45
BOSSNET	100	100	95.2	100	97.3	100	100	95.7	100	91.7	15.2	43.1

Table 6: Ablation study: impact of each model element on BOSSNET

Most of the existing end-to-end approaches *retrieve* a response from a pre-defined set (Bordes and Weston, 2017; Liu and Perez, 2017; Seo et al., 2017). These methods are generally successful when they have to provide boilerplate responses – they cannot construct responses by using words in KB not seen during training. Alternatively, generative approaches are used where the response is *generated* one word at a time (Eric and Manning, 2017; Madotto et al., 2018). These approaches mitigate the unseen entity problem by incorporating the ability to copy words from the input (Vinyals et al., 2015; Gu et al., 2016). The copy mechanism has also found success in summarization (Nallapati et al., 2016; See et al., 2017) and machine translation (Gulcehre et al., 2016). BOSSNET is also a copy incorporated generative approach.

For encoding, some approaches represent the dialog history as a sequence (Eric and Manning, 2017; Gulcehre et al., 2016). Unfortunately, using a single long sequence for encoding also enforces an order over the set of KB tuples making it harder to perform inferencing over them. Other approaches represent the dialog context as a bag. Original Memory Networks (Bordes and Weston, 2017) and its extensions encode each memory element (utterance) as an average of all constituent words – this cannot point to individual words, and hence cannot be used with a copy mechanism. Mem2Seq encodes each word individually in a flat memory. Unfortunately, this loses the contextual information around a word, which is needed to decipher an unseen word. In contrast, BOSSNET uses a bag of sequences encoding, where KB tuples are a set for easier inference, and also each utterance is a sequence for effectively learning when to copy.

KB (restaurant—cuisine—address—phone)	
pizza_hut_fen_ditton—italian—cambridge_retail_park_newmarket_road_fen_ditton—01223_323737	

usr-1	may i have information for an *italian* restaurant in the *east* part of town ?
sys-1	yes sure . there is only on *italian* restaurant called *pizza_hut_fen_ditton* in the *east* . what else do you want to know ?
usr-2	what is their address please ?
Gold	the address for *pizza_hut_fen_ditton* is *cambridge_retail_park_newmarket_road_fen_ditton* . can i help you with anything else ?
Seq2Seq+Copy	their address is at
Seq2Seq	*curry_prince* is at *451_newmarket_road_fen_ditton*
Mem2Seq	*pizza_hut_fen_ditton* is located at *7_barnwell_road_fen_ditton* is *7_barnwell_road_fen_ditton* and the phone number is *01223_244955*
BOSSNET	*pizza_hut_fen_ditton* is located at *cambridge_retail_park_newmarket_road_fen_ditton* is there anything else i can help you with

Table 7: An example of responses generated by BOSSNET and baselines on the CamRest test set. Thia example has no unseen entities.

KB (restaurant—cuisine—price_range—location—rating)	
resto_tokyo_affordable_vietnamese_8stars—vietnamese—affordable—tokyo—8	
resto_tokyo_affordable_vietnamese_7stars—vietnamese—affordable—tokyo—7	
resto_tokyo_affordable_vietnamese_6stars—vietnamese—affordable—tokyo—6	
resto_tokyo_affordable_vietnamese_5stars—vietnamese—affordable—tokyo—5	

usr-1	can you book a table in *tokyo* with *vietnamese* food in a *affordable* price range ?
sys-1	what do you think of this option : *resto_tokyo_affordable_vietnamese_8stars* ?
usr-2	no this does not work for me .
sys-2	what do you think of this option : *resto_tokyo_affordable_vietnamese_7stars* ?
usr-3	do you have something else ?
Gold	what do you think of this option : *resto_tokyo_affordable_vietnamese_6stars*
Seq2Seq+Copy	what do you think of this option : what ?
Seq2Seq	what do you think of this option : *resto_london_moderate_british_2stars* ?
Mem2Seq	what do you think of this option : *resto_tokyo_affordable_vietnamese_5stars* ?
BOSSNET	what do you think of this option : *resto_tokyo_affordable_vietnamese_6stars* ?

Table 8: An example of responses generated by BOSSNET and baselines on bAbI dialog Task-5. This example is from the KA test set with 100% unseen entities.

6   Conclusions

We propose BOSSNET for training task-oriented dialog systems in an end-to-end fashion. BOSS-NET combines a novel bag of sequences memory for storing a dialog history and KB tuples, with a copy-augmented generative decoder to construct dialog responses. It augments standard cross entropy loss of a sequence decoder with an additional term to encourage the model to copy KB words. BOSS memory and new loss term, in conjunction with a disentangle label dropout, enables the decoder to disentangle its language and knowledge models.

BOSSNET achieves the state of the art results on bAbI dialog dataset, outperforming existing models by 10 points or more in its OOV conditions. In the knowledge adaptability test, we find that BOSSNET is highly robust to increasing the per-centage of unseen entities at test time, suggesting a good language-knowledge disentanglement. Human evaluations show that BOSSNET responses are highly informative and slightly more grammatical compared to baselines. We will release our code and all curated datasets 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, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Dialog System Technology Challenges, Saarbrücken, Germany, August 15-17, 2017*, pages 37–49.

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.

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 Steve Young. 2014. 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*.

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.

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.

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*.

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

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.

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. ACL.

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.

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

A Two-Level attention on BoSs Memory

To visualize the benefit of two-level attention used on BOSS memory by the decoder, we compare attention weights for two models: our proposed *two-level attention* and a variant with just *one-level attention* (over all the words in the memory). In the example of a sample dialog from bAbI Task 3, shown in Figure 7, the decoder is 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 g_s values, 3.14 x 10^{-5} and 1.15 x 10^{-4} for two-level attention and one-level attention 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 two-level attention 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 (*3 stars > 2 stars > 1 star*) and suppresses the weights of any previously suggested restaurant.

The attention over memory cells provides BOSS-NET 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 Memory cells. Attention over tokens on the other hand can push the attention weights towards either the subject or object in the KB tuple, based on the query's request. Thus using both in conjunction helps BOSSNET perform significantly better than the baselines and illustrates the importance of the BOSS memory in comparison to a flat memory layout.

B Reproducibility

We list out the complete set of hyperparameters used to train BOSSNET for the various datasets in Table 9. Our code will be made publicably accessible for future research purposes. Our trained models and evaluation scripts will also be provided. We will also make our end-to-end reconstruced Camrest dataset along with our whole batch of knowledge adaptability test sets available.

C Example Predictions of BOSSNET and Baselines

Examples from SMD is shown in Table 12 respectively. Examples from KA test set with percentage of unseen entites set to 50 from CamRest and SMD are shown in Table 11 and Table 13 respectively. Examples from KA test set with percentage of unseen entites set to 100 from bAbI dialog Task 1 is shown in Table 10.

D Dataset Preprocessing and Faults

D.1 Mem2Seq Preprocessing

Mem2Seq paper used the following pre-processing on the data:

1. The subject (restaurant name) and object (rating) positions of the rating KB tuples in bAbI dialogs are flipped, while the order remains the same for other tuples remains the same. This pre-processing is illustrated in Figure 8

2. an extra fact was added to the navigation tasks in In-Car Assistant with all the properties (such as distance, address) combined together as the subject and *poi* as the object. This pre-processing is illustrated in Figure 9

The pre-processing has major impact on the performance of Mem2Seq, as it can only copy objects of a KB tuple, while the subject and relation can never be copied.

D.2 bAbI Dataset Faults

The KB entities present in validation and non-OOV test sets for task 3 and 4 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. This explains the low performance of baseline models on task 3 and 4 non-OOV test sets.

E AMT Setup

Response Relevance Test We show a sample of an Human Intelligence Task (HIT) on Amazon Mechanical Turk in Figure 10a. We randomize the responses generated by the three baseline models and BOSSNET on the same dialog and ask the user to tick all those response options that seem to capture the relevant information of the given sample response. A total of 200 such annotations were collected for Camrest and SMD each.
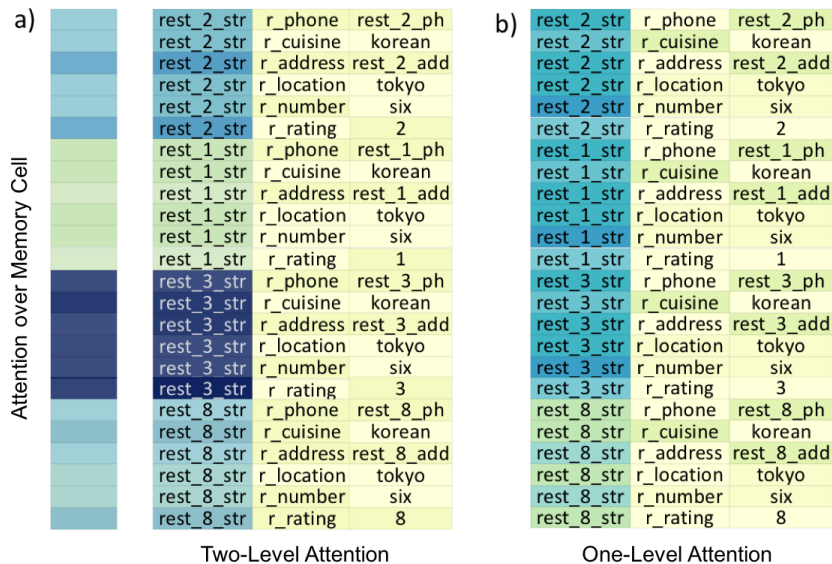
Figure 7: Visualization of attention weights on selected portions of memory in (a) BOSSNET with two-level attention vs (b) BOSSNET with one-level attention

Task	Learning Rate	Hops	Embedding Size	Disentangle Loss Weight	DLD
T1	0.001	1	128	1.0	0.2
T2	0.001	1	128	1.0	0.2
T3	0.0005	3	128	1.5	0.2
T4	0.001	1	128	1.0	0.2
T5	0.0005	3	256	1.0	0.2
CamRest	0.0005	6	256	1.0	0.2
SMD	0.0005	3	256	1.0	0.1

Table 9: The hyperparameters used to train BOSSNET on the different datasets
.

Response Grammar Test We show a sample of an Human Intelligence Task (HIT) on Amazon Mechanical Turk in Figure 10b. We randomize the responses generated by the three baseline models and BOSSNET on the same dialog and ask the user to rate each response based on the grammatical correctness and natural flow of the sentence. The rating ranges from 0-3 where 0 being the worst and 3 being the best. Note) the sentences were not asked to be rated with respect to each other, but instead as individual occurrences. A total of 200 such annotations were collected for Camrest and SMD each.

F Multi-Hop vs 1-Hop Encoders

Table 14 shows the performance of bAbI tasks and CamRest on two BOSSNET encoder settings. Multi-hops in encoder helps in bAbI task 3 and 5, as they require inferencing over the KB tuples (sorting restaurants by rating) to recommend a restaurant. We also see substantial improvements on CamRest in both BLEU and entity F1 metric.

usr-1	good morning
sys-1	hello what can i help you with today
usr-2	can you book a table with *korean* cuisine for *five* in *beijing*
sys-2	i ' m on it . which price range are looking for ?
usr-3	i am looking for a *overpriced* restaurant

Gold	api_call *korean beijing five overpriced*

Seq2Seq+Copy	api_call *korean beijing five overpriced*

Seq2Seq	api_call *french london four moderate*

Mem2Seq	api_call *overpriced korean korean overpriced*

BOSSNET	api_call *korean beijing five overpriced*

Table 10: Example from bAbI dialog Task 1 with 100% OOV.

KB (restaurant—cuisine—address—phone—postcode)
prezzo—ethiopian—22_atlantis_road—01799_521260—cb30ad

usr-1	i want a *moderately_expensive ethiopian* restaurant in the *south* side of town
sys-1	i ' m sorry . i can not find a *moderately_expensive ethiopian* restaurant in the *south* side of town
usr-2	i would like to find a *moderately* priced restaurant in the *west* part of town
sys-2	*prezzo* is a moderately priced *ethiopian* restaurant in the *west* part of town . there is also *indian* and *british* food available in that area and price range . would you like more info ?
usr-3	yes can i get thier address and postcode ?

Gold	of course , *prezzo* ' s address is *22_atlantis_road* and postcode is *cb30ad*

Seq2Seq+Copy	postcode a is is *01223350688* UNK there get i for for today

Seq2Seq	the address is *83_regent_street* and their postcode is *cb58aq*

Mem2Seq	the phone address is *cb21uj*

BOSSNET	the address is *22_atlantis_road* at *22_atlantis_road* the postcode is *cb30ad* is there anything else i can help you with

Table 11: Example from Camrest with 50% OOV. The OOV entities present in the dialog are {*ethiopian*, *22_atlantis_road*}

KB (location—distance—traffic info—address)
teavana—4_miles—no_traffic—145_amherst_st
cafe_venetia—5_miles—moderate_traffic—269_alger_dr
town_and_country—3_miles—moderate_traffic—383_university_ave

usr-1	where can i get tea ?
sys-1	there is 2 tea shops near you *teavana* is *4_miles* away and *cafe_venetia* is *5_miles* away which one do you want ?
usr-2	which one is the quickest is there any traffic ?

Gold	*teavana* is the fastest journey because there is *no_traffic* on the way there . *cafe_venetia* has *moderate_traffic* should we go to *teavana*

Seq2Seq+Copy	there is

Seq2Seq	there is *moderate_traffic* on the way to *palo_alto_medical_foundation*

Mem2Seq	there is *moderate_traffic* on our way

BOSSNET	*teavana* there is a *moderate_traffic* nearby but i might find another route to *town_and_country town_and_country* address is *383_university_ave* i sent it on your screen

Table 12: Example from SMD

	KB (location—day—weather)
	mountain_view—sunday—foggy
	mountain_view—monday—overcast
	pittsburgh—sunday—overcast
	pittsburgh—monday—warm
	downtown_chicago—sunday—foggy
	downtown_chicago—monday—warm

today—monday

usr-1	is the weather *cloudy* right now ?
sys-1	what city do you want the weather for ?
usr-2	*pittsburgh*
Gold	there are no clouds in *pittsburgh* right now
Seq2Seq+Copy	which is no
Seq2Seq	it is currently raining in *san_francisco* right now
Mem2Seq	it will not be cloudy in in
BOSSNET	it will be *cloudy* on *sunday* in *pittsburgh*

Table 13: Example from SMD with 50% OOV. The OOV entity present in the dialog is {*pittsburgh*}

	bAbI Dialog Tasks					bAbI Dialog Tasks (OOV)					CamRest	
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	BLEU	Ent. F1
BOSSNET with 1-Hop Encoder	100	100	92.3	100	90.5	100	100	91.4	100	89	10.5	36.9
BOSSNET with Multi-Hop Encoder	100	100	95.2	100	97.3	100	100	95.7	100	91.7	15.2	43.1

Table 14: Ablation study: impact of hops in BOSSNET encoder

(a) Original bAbI Data

Subject	Predicate	Object
resto_rome_cheap_indian_6stars	R_phone	resto_rome_cheap_indian_6stars_phone
resto_rome_cheap_indian_6stars	R_cuisine	indian
resto_rome_cheap_indian_6stars	R_address	resto_rome_cheap_indian_6stars_address
resto_rome_cheap_indian_6stars	R_location	rome
resto_rome_cheap_indian_6stars	R_number	eight
resto_rome_cheap_indian_6stars	R_price	cheap
resto_rome_cheap_indian_6stars	R_rating	6
resto_rome_cheap_indian_7stars	R_phone	resto_rome_cheap_indian_7stars_phone
resto_rome_cheap_indian_7stars	R_cuisine	indian
resto_rome_cheap_indian_7stars	R_address	resto_rome_cheap_indian_7stars_address
resto_rome_cheap_indian_7stars	R_location	rome
resto_rome_cheap_indian_7stars	R_number	eight
resto_rome_cheap_indian_7stars	R_price	cheap
resto_rome_cheap_indian_7stars	R_rating	7

⬇

Subject	Predicate	Object
resto_rome_cheap_indian_6stars	R_phone	resto_rome_cheap_indian_6stars_phone
resto_rome_cheap_indian_6stars	R_cuisine	indian
resto_rome_cheap_indian_6stars	R_address	resto_rome_cheap_indian_6stars_address
resto_rome_cheap_indian_6stars	R_location	rome
resto_rome_cheap_indian_6stars	R_number	eight
resto_rome_cheap_indian_6stars	R_price	cheap
6	R_rating	resto_rome_cheap_indian_6stars
resto_rome_cheap_indian_7stars	R_phone	resto_rome_cheap_indian_7stars_phone
resto_rome_cheap_indian_7stars	R_cuisine	indian
resto_rome_cheap_indian_7stars	R_address	resto_rome_cheap_indian_7stars_address
resto_rome_cheap_indian_7stars	R_location	rome
resto_rome_cheap_indian_7stars	R_number	eight
resto_rome_cheap_indian_7stars	R_price	cheap
7	R_rating	resto_rome_cheap_indian_7stars

(a) Pre-Processed bAbI Data

Figure 8: Pre-processing of bAbI dialog data used in Mem2Seq paper

(a) Original SMD Navigate Data

Subject	Predicate	Object
the_westin	distance	2_miles
the_westin	traffic_info	moderate_traffic
the_westin	poi_type	rest_stop
the_westin	address	329_el_camino_real
toms_house	distance	1_miles
toms_house	traffic_info	heavy_traffic
toms_house	poi_type	friends_house
toms_house	address	580_van_ness_ave

Subject	Predicate	Object
2_miles moderate_traffic rest_stop	poi	the_westin
the_westin	distance	2_miles
the_westin	traffic_info	moderate_traffic
the_westin	poi_type	rest_stop
the_westin	address	329_el_camino_real
1_miles heavy_traffic friends_house	poi	toms_house
toms_house	distance	1_miles
toms_house	traffic_info	heavy_traffic
toms_house	poi_type	friends_house
toms_house	address	580_van_ness_ave

(a) Pre-Processed SMD Navigate Data

Figure 9: Pre-processing of SMD Navigate data used in Mem2Seq paper

Chatbot Response Validation (Click to expand)

Each HIT contains an incomplete conversation between a human and a bot that assists in reserving a table in restaurants. The list of restaurants the bot has access to is listed below.

Please select all the options that could be a relevant response to the incomplete conversation. A sample relevant response is posted below.

If none of the responses are appropriate, then please select the best one of the four

If an option conveys the relevant information, ignoring the grammatical mistakes, then please select the option.

Restaurant List

Restaurant	Address	Area	Cuisine	Phone	Pricerange	Postcode
da_vinci_pizzeria	20_milton_road_chesterton	north	italian	01223_351707	cheap	cb41jy

Conversation So Far:

USER	i am interested in a restaurant located in the north part of town that specializes in indonesian food , please
BOT	there are no restaurants serving indonesian food in the north part of town . would you like me to look in another part of town
USER	what about italian food
BOT	da_vinci_pizzeria is a restaurant serving italian food in the north part of town . would you like any additional information
USER	yes , may i have the address and phone number please

Sample Valid Respose: the address is 20_milton_road_chesterton and the phone number is 01223_351707

Option_1	the address is 20_milton_road_chesterton at 20_milton_road_chesterton the phone number is 01223_351707 is there anything else i can help you with
Option_2	da_vinci_pizzeria and the phone number
Option_3	the address is 20_milton_road_chesterton , and phone number is 01223_351707
Option_4	sure is at 30_bridge_street_city_centre and their phone number is is UNK there anything i i i help you you

Select all the options that are possible valid responses to the above conversation:
- ☐ Option_1
- ☐ Option_2
- ☐ Option_3
- ☐ Option_4

You must ACCEPT the HIT before you can submit the results.

(a)

Instructions

We present 4 different responses that have been created by different chatbots:

- Please rate each response based on grammatical correctness and fluency.
- Rating will be done on a scale of 0-3, where 0 is a completely incorrect sentence and 3 is a fully/nearly grammatically correct and natural sounding sentence.
- Note) The sentences are responses to specific tasks, so don't penalize for proper nouns like "flying_duck" which could be a possible restaurant suggested to the user.
- If a certain response is empty, give it a 0 rating.
- Give a low rating if the sentence is incomplete, e.g., "the weather"

Example
- the the the is restaurant in in in . . . **Rating 1**
- the flying_duck is is a restaurant in the south part of town. **Rating 3**
- the flying_duck is a . part of town **Rating 2**
- . . . restaurant is in . of town **Rating 1**

Option_1	frankie_and_bennys is a italian restaurant in the south . would you like their phone number
Option_2	there is an expensive restaurant in the the the the the the the the the the the
Option_3	chiquito_restaurant_bar is an expensive restaurant in the the part part of of town would you like their their
Option_4	is expensive expensive the south south

Select all the options that are possible valid responses to the above conversation:

Option 1
- select rating -

Option 2
- select rating -

Option 3
- select rating -

Option 4
- select rating -

You must ACCEPT the HIT before you can submit the results.

(b)

Figure 10: A sample HIT on Amazon Mechanical Turk to (a) validate useful responses based on the given dialog context, and (b) validate grammatical correctness of different responses on a scale of 0-3