Structured Fusion Networks for Dialog

Shikib Mehri*, **Tejas Srinivasan***, **and Maxine Eskenazi**
Language Technologies Institute, Carnegie Mellon University
{amehri,tsriniva,max+}@cs.cmu.edu

Abstract

Neural dialog models have exhibited strong performance, however their end-to-end nature lacks a representation of the explicit structure of dialog. This results in a loss of generalizability, controllability and a data-hungry nature. Conversely, more traditional dialog systems do have strong models of explicit structure. This paper introduces several approaches for explicitly incorporating structure into neural models of dialog. Structured Fusion Networks first learn neural dialog modules corresponding to the structured components of traditional dialog systems and then incorporate these modules in a higher-level generative model. Structured Fusion Networks obtain strong results on the MultiWOZ dataset, both with and without reinforcement learning. Structured Fusion Networks are shown to have several valuable properties, including better domain generalizability, improved performance in reduced data scenarios and robustness to divergence during reinforcement learning.

1 Introduction

End-to-end neural dialog systems have shown strong performance (Vinyals and Le, 2015; Dinan et al., 2019). However such models suffer from a variety of shortcomings, including: a data-hungry nature (Zhao and Eskenazi, 2018), a tendency to produce generic responses (Li et al., 2016b), an inability to generalize (Mo et al., 2018; Zhao and Eskenazi, 2018), a lack of controllability (Hu et al., 2017), and divergent behavior when tuned with reinforcement learning (Lewis et al., 2017). Traditional dialog systems, which are generally free of these problems, consist of three distinct components: the natural language understanding (NLU), which produces a structured representation of an input (e.g., a belief state); the natural language generation (NLG), which produces output in natural language conditioned on an internal state (e.g. dialog acts); and the dialog manager (DM) (Bohus and Rudnicky, 2009), which describes a policy that combines an input representation (e.g., a belief state) and information from some database to determine the desired continuation of the dialog (e.g., dialog acts). A traditional dialog system, consisting of an NLU, DM and NLG, is pictured in Figure 1.

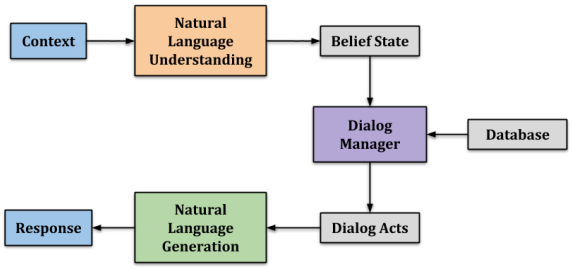


Figure 1: A traditional dialog system consisting of a natural language understanding (NLU), dialog manager (DM) and natural language generation (NLG).

The structured components of traditional dialog systems facilitate effective generalizability, interpretability, and controllability. The structured output of each component allows for straightforward modification, understanding and tuning of the system. On the other hand, end-to-end neural models of dialog lack an explicit structure and are treated as a black box. To this end, we explore several methods of incorporating the structure of traditional dialog systems into neural dialog models.

First, several neural *dialog modules* are constructed to serve the role of the NLU, the DM and the NLG. Next, a number of methods are proposed for incorporating these dialog modules into end-to-end dialog systems, including Naïve Fusion, Multitask Fusion and Structured Fusion Networks (SFNs). This paper will show that SFNs

* Equal contribution.

Proceedings of the SIGDial 2019 Conference, pages 165–177
Stockholm, Sweden, 11-13 September 2019.

obtain strong results on the MultiWOZ dataset (Budzianowski et al., 2018) both with and without the use of reinforcement learning. Due to the explicit structure of the model, SFNs are shown to exhibit several valuable properties including improved performance in reduced data scenarios, better domain generalizability and robustness to divergence during reinforcement learning (Lewis et al., 2017).

2 Related Work

2.1 Generation Methods

Vinyals and Le (2015) used a sequence-to-sequence network (Sutskever et al., 2014) for dialog by encoding the conversational context and subsequently generating the reply. They trained and evaluated their model on the OpenSubtitles dataset (Tiedemann, 2009), which contains conversations from movies, with a total of 62M training sentences.

Most research on generative models of dialog has built on the baseline introduced by Vinyals and Le (2015) by incorporating various forms of inductive bias (Mitchell, 1980) into their models, whether it be through the training procedure, the data or through the model architecture. Li et al. (2015) use Maximum Mutual Information (MMI) as the objective function, as a way of encouraging informative agent responses. Serban et al. (2016) proposes to better capture the semantics of dialog with the use of a hierarchical encoder decoder (HRED), comprised of an utterance encoder, a conversational context encoder, and a decoder. Li et al. (2016b) incorporate a number of heuristics into the reward function, to encourage useful conversational properties such as informativity, coherence and forward-looking. Li et al. (2016a) encodes a speaker's persona as a distributed embedding and uses it to improve dialog generation. Liu and Lane (2016) simultaneously learn intent modelling, slot filling and language modelling. Zhao et al. (2017) enables task-oriented systems to make slot-value-independent decisions and improves out-of-domain recovery through the use of entity indexing and delexicalization. Wu et al. (2017) present Recurrent Entity Networks which use action templates and reasons about abstract entities in an end-to-end manner. Zhao and Eskenazi (2018) present the Action Matching algorithm, which maps utterances to a cross-domain embedding space to improve zero-shot generalizability. Mehri et al. (2019) explore several dialog specific pre-training objectives that improve performance on dowstrean dialog tasks, including generation. Chen et al. (2019) present a hierarchical self-attention network, conditioned on graph structured dialog acts and pre-trained with BERT (Devlin et al., 2018).

2.2 Generation Problems

Despite their relative success, end-to-end neural dialog systems have been shown to suffer from a number of shortcomings. (Li et al., 2016b) introduced the dull response problem, which describes how neural dialog systems tend to produce generic and uninformative responses (e.g., *"I don't know"*). Zhao and Eskenazi (2018) describe generative dialog models as being data-hungry, and difficult to train in low-resource environments. Mo et al. (2018); Zhao and Eskenazi (2018) both demonstrate that dialog systems have difficulty generalizing to new domains. Hu et al. (2017) work on the problem of controllable text generation, which is difficult in sequence-to-sequence architectures, including generative models of dialog.

Wang et al. (2016) describe the problem of the *overwhelming implicit language model* in image captioning model decoders. They state that the decoder learns a language generation model along with a policy, however, during the process of captioning certain inputs, the decoder's implicit language model overwhelms the policy and, as such, generates a specific output regardless of the input (e.g., if it generates 'giraffe', it may always output 'a giraffe standing in a field', regardless of the image). In dialog modelling, this problem is observed in the output of dialog models fine-tuned with reinforcement learning (Lewis et al., 2017; Zhao et al., 2019). Using reinforcement learning to fine-tune a decoder, will likely place a strong emphasis on improving the decoder's policy and un-learn the implicit language model of the decoder. To this end, Zhao et al. (2019) proposes Latent Action Reinforcement Learning which does not update the decoder during reinforcement learning.

The methods proposed in this paper aim to mitigate these issues by explicitly modelling structure. Particularly interesting is that the structured models will reduce the effect of the *overwhelming implicit language model* by explicitly modelling the

NLG (i.e., a conditioned language model). This should lessen the divergent effect of reinforcement learning (Lewis et al., 2017; Zhao et al., 2019).

2.3 Fusion Methods

This paper aims to incorporate several pre-trained *dialog modules* into a neural dialog model. A closely related branch of research is the work done on fusion methods, which attempts to integrate pre-trained language models into sequence-to-sequence networks. Integrating language models in this manner is a form of incorporating structure into neural architectures. The simplest such method, commonly referred to as **Shallow Fusion**, is to add a language modeling term, $p_{LM}(y)$, to the cost function during inference (Chorowski and Jaitly, 2016).

To improve on this, Gulcehre et al. (2015) proposed **Deep Fusion**, which combines the states of a pre-trained machine translation models decoder and a pre-trained language model by concatenating them using a gating mechanism with trained parameters. The gating mechanism allows us to decide how important the language model and decoder states are at each time step in the inference process. However, one major drawback of Deep Fusion is that the sequence-to-sequence model is trained independently from the language model, and has to learn an implicit language model from the training data.

Cold Fusion (Sriram et al., 2017) deals with this problem by training the sequence-to-sequence model along with the gating mechanism, thus making the model aware of the pre-trained language model throughout the training process. The decoder does not need to learn a language model from scratch, and can thus learn more task-specific language characteristics which are not captured by the pre-trained language model (which has been trained on a much larger, domain-agnostic corpus).

3 Methods

This section describes the methods employed in the task of dialog response generation. In addition to the baseline model proposed by Budzianowski et al. (2018), several methods of incorporating structure into end-to-end neural dialog models are explored.

3.1 Sequence-to-Sequence

The baseline model for dialog generation, depicted in Figure 2, consists of a standard encoder-decoder framework (Sutskever et al., 2014), augmented with a belief tracker (obtained from the annotations of the dialog state) and a database vector. The dialog system is tasked with producing the appropriate system response, given a dialog context, an oracle belief state representation and a vector corresponding to the database output.

The dialog context is encoded using an LSTM (Hochreiter and Schmidhuber, 1997) sequence-to-sequence network (Sutskever et al., 2014). Experiments are conducted with and without an attention mechanism (Bahdanau et al., 2015). Given the final encoder hidden state, h_t^e, the belief state vector, v_{bs}, and the database vector, v_{db}, Equation 1 describes how the initial decoder hidden state is obtained.

$$h_0^d = tanh\left(W_e h_t^e + W_{bs} v_{bs} + W_{db} v_{db} + b\right) \quad (1)$$

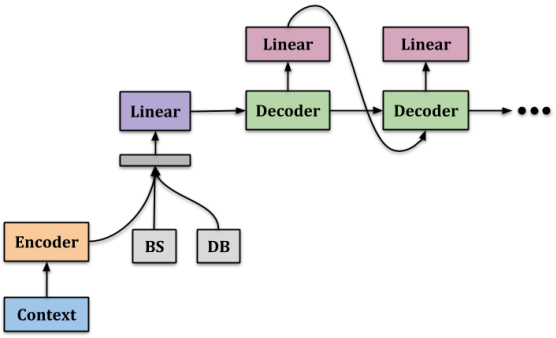


Figure 2: A diagram of the baseline sequence-to-sequence architecture. The attention mechanism is not visualized, however experiments are conducted both with and without attention.

3.2 Neural Dialog Modules

As seen in Figure 1, a traditional dialog system consists of the NLU, the DM and the NLG. The NLU maps a natural language input to a belief state representation (BS). The DM uses the belief state and some database output, to produce dialog acts (DA) for the system response. The NLG uses the dialog acts to produce a natural language response.

A neural *dialog module* is constructed for each of these three components. A visualization of these architectures is shown in Figure 3. The NLU architecture uses an LSTM encoder to map the

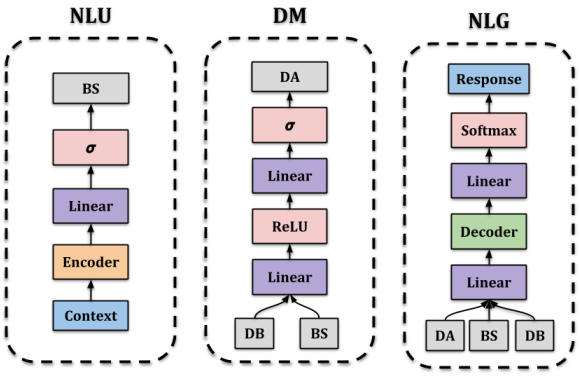


Figure 3: A visualization of the neural architectures for each of the three modules of traditional dialog systems.

natural language input to a latent representation, h_t, which is then passed through a linear layer and a sigmoid function to obtain a multi-label prediction of the belief state. The DM architecture projects the belief state and database vector into a latent space, through the use of a linear layer with a ReLU activation, which is then passed through another linear layer and a sigmoid function to predict the dialog act vector. The neural architecture corresponding to the NLG is a conditioned language model with its initial hidden state given by a linear encoding of the dialog acts, belief state and database vectors.

The following equations define the structure of the modules, where the gt subscript on an intermediate variable denotes the use of the ground-truth value:

$$bs = \textbf{NLU}(\texttt{context}) \quad (2)$$

$$da = \textbf{DM}(bs_{gt}, db) \quad (3)$$

$$\texttt{response} = \textbf{NLG}(bs_{gt}, db, da_{gt}) \quad (4)$$

3.3 Naïve Fusion

Naïve Fusion (NF) is a straightforward mechanism for using the neural dialog modules for end-to-end dialog response generation.

3.3.1 Zero-Shot Naïve Fusion

During training, each dialog module is trained independently, meaning that it is given the ground truth input and supervision signal. However, during inference, the intermediate values (e.g., the dialog act vector) do not necessarily exist and the outputs of other neural modules must be used instead. For example, the DM module is trained given the ground-truth belief state as input, however during inference it must rely on the belief state predicted by the NLU module. This results in a propagation of errors, as the DM and NLG may receive imperfect input.

Zero-Shot Naïve Fusion combines the pre-trained neural modules at inference time. The construction of the response conditioned on the context, is described as follows:

$$bs = \textbf{NLU}(\texttt{context}) \quad (5)$$

$$\texttt{response} = \textbf{NLG}(bs, db, \textbf{DM}(bs, db)) \quad (6)$$

3.3.2 Naïve Fusion with Fine-Tuning

Since the forward propagation described in Equations 5 and 6 is continuous and there is no sampling procedure until the response is generated, Naïve Fusion can be fine-tuned for the end-to-end task of dialog generation. The pre-trained neural modules are combined as described above, and fine-tuned on the task of dialog generation using the same data and learning objective as the baseline.

3.4 Multitask Fusion

Structure can be incorporated into neural architectures through the use of multi-tasking. Multitask Fusion (MF) is a method where the end-to-end generation task is learned simultaneously with the aforementioned dialog modules. The multitasking setup is seen in Figure 4.

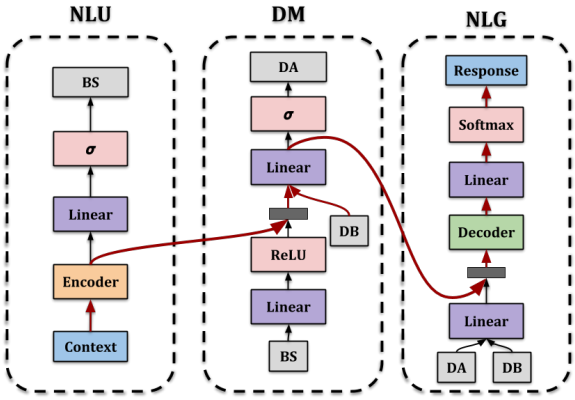


Figure 4: A depiction of Multitask Fusion, where the individual neural modules are learned simultaneously with the end-to-end task of dialog generation. The dashed boxes contain the individual components, while the red arrows depict forward propagation for the end-to-end task. The red arrows are the process used during response generation.

By sharing the weights of the end-to-end architecture and each respective module, the learned representations should become stronger and more structured in nature. For example, the encoder is

shared between the NLU module and the end-to-end task. As such, it will learn to both represent the information necessary for predicting the belief state vector and any additional information useful for generating the next utterance.

3.5 Structured Fusion Networks

The Structured Fusion Networks (SFNs) we propose, depicted in Figure 5, use the independently pre-trained neural dialog modules for the task of end-to-end dialog generation. Rather than fine-tuning or multi-tasking the independent modules, SFNs aim to learn a higher-level model on top of the neural modules to perform the task of end-to-end response generation.

The output of the NLU is concatenated at each time-step of the encoder input. The output of the DM is similarly concatenated to the input of the linear layer between the encoder and the decoder of the higher-level model. The output of the NLG, in the form of logits at a decoding time-step, is combined with the hidden state of the decoder via cold-fusion (Sriram et al., 2017). Given the NLG output as l_t^{NLG} and the higher-level decoder hidden state as s_t, the cold-fusion method is described as follows:

$$h_t^{NLG} = DNN(l_t^{NLG}) \quad (7)$$

$$g_t = \sigma(W[s_t; h_t^{NLG}] + b) \quad (8)$$

$$s_t^{CF} = [s_t; g_t \circ h_t^{NLG}] \quad (9)$$

$$y_t = softmax(DNN(s_t^{CF})) \quad (10)$$

By pre-training the modules and using their structured outputs, the higher-level model does not have to *re-learn* and *re-model* the dialog structure (i.e., representing the belief state and dialog acts). Instead, it can focus on the more abstract modelling that is necessary for the task, including recognizing and encoding complex natural language input, modelling a policy, and effectively converting a latent representation into a natural language output according to the policy.

The SFN architecture may seem complicated due to the redundancy of the inputs. For example, the context is passed to the model in two places and the database vector in three places. This redundancy is necessary for two reasons. First, each of the neural modules must function independently and thus needs sufficient inputs. Second, the higher-level model should be able to function well independently. If any of the neural modules was to be removed, the SFN should be able to perform reasonably. This means that the higher-level module should not rely on any of the neural modules to capture information about the input and therefore allow the neural modules to focus only on representing the structure. For example, if the context was not passed into the higher-level encoder and instead only to the NLU module, then the NLU may no longer be able to sufficiently model the belief state and may instead have to more explicitly model the context (e.g., as a bag-of-words representation).

Several variations of training SFNs are considered during experimentation, enumerated as follows. (1) The pre-trained neural modules are kept frozen, as a way of ensuring that the structure is not deteriorated. (2) The pre-trained neural modules are fine-tuned for the end-to-end task of response generation. This ensures that the model is able to abandon or modify certain elements of the structure if it helps with the end-to-end task. (3) The pre-trained modules are multi-tasked with the end-to-end task of response generation. This ensures that the structure is maintained and potentially strengthened while also allowing the modules to update and improve for the end-to-end task.

4 Experiments

4.1 Dataset

The dialog systems are evaluated on the MultiWOZ dataset (Budzianowski et al., 2018), which consists of ten thousand human-human conversations covering several domains. The MultiWOZ dataset contains conversations between a tourist and a clerk at an information center which fall into one of seven domains - attraction, hospital, police, hotel, restaurant, taxi, train. Individual conversations span one to five of the domains. Dialogs were collected using the Wizard-of-Oz framework, where one participant plays the role of an automated system.

Each dialog consists of a goal and multiple user and system utterances. Each turn is annotated with two binary vectors: a belief state vector and a dialog act vector. A single turn may have multiple positive values in both the belief state and dialog act vectors. The belief state and dialog act vectors are of dimensions 94 and 593, respectively.

Several metrics are used to evaluate the models. BLEU (Papineni et al., 2002) is used to com-

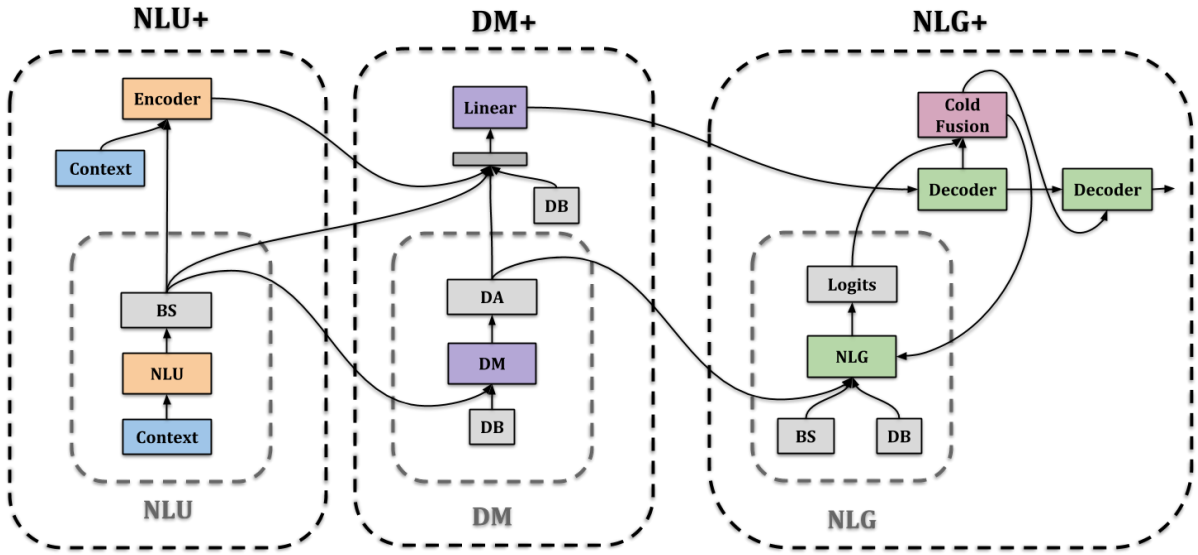


Figure 5: The Structured Fusion Network. The grey dashed boxes correspond to the pre-trained neural dialog modules. A higher-level is learned on top of the pre-trained modules, as a mechanism of enforcing structure in the end-to-end model.

pute the word overlap between the generated output and the reference response. Two task-specific metrics, defined by Budzianowski et al. (2018), Inform rate and Success rate, are also used. Inform rate measures how often the system has provided the appropriate entities to the user. Success rate measures how often the system answers all the requested attributes. Similarly to Budzianowski et al. (2018), the best model is selected during validation using the combined score which is defined as $BLEU + 0.5 \times (Inform + Success)$. This combined score is also reported as an evaluation metric.

4.2 Experimental Settings

The hyperparameters match those used by Budzianowski et al. (2018): embedding dimension of 50, hidden dimension of 150, and a single-layer LSTM. All models are trained for 20 epochs using the Adam optimizer (Kingma and Ba, 2014), with a learning rate of 0.005 and batch size of 64. The norm of the gradients are clipped to 5 (Pascanu et al., 2012). Greedy decoding is used during inference.

All previous work uses the ground-truth belief state vector during training and evaluation. Therefore the experiments with the SFNs have the NLU module replaced by an "oracle NLU" which always outputs the ground-truth belief state. Table 4 in the Appendix shows experimental results which demonstrate that using only the ground-truth belief state results in the best performance.

4.3 Reinforcement Learning

A motivation of explicit structure is the hypothesis that it will reduce the effects of the implicit language model, and therefore mitigate degenerate output after reinforcement learning. This hypothesis is evaluated by fine-tuning the SFNs with reinforcement learning. The setup for this experiment is similar to that of Zhao et al. (2019): (1) the model produces a response conditioned on a ground-truth dialog context, (2) the success rate is evaluated for the generated response, (3) using the success rate as the reward, the policy gradient is calculated at each word, and (4) the parameters of the model are updated. A learning rate of `1e-5` is used with the Adam optimizer (Kingma and Ba, 2015).

Reinforcement learning is used to fine-tune the best performing model trained in a supervised learning setting. During this fine-tuning, the neural dialog modules (i.e., the NLU, DM and NLG) are frozen. Only the high-level model is updated during reinforcement learning. Freezing maintains the structure, while still updating the higher level components. Since the structure is maintained, it is unnecessary to alternate between supervised and reinforcement learning.

Model	BLEU	Inform	Success	Combined Score
Supervised Learning				
Seq2Seq (Budzianowski et al., 2018)	18.80	71.29%	60.29%	84.59
Seq2Seq w/ Attn (Budzianowski et al., 2018)	18.90	71.33%	60.96%	85.05
Seq2Seq (Ours)	20.78	61.40%	54.50%	78.73
Seq2Seq w/ Attn (ours)	20.36	66.50%	59.50%	83.36
3-layer HDSA (Chen et al., 2019)	**23.60**	**82.90%**	**68.90%**	**99.50**
Naïve Fusion (Zero-Shot)	7.55	70.30%	36.10%	60.75
Naïve Fusion (Fine-tuned Modules)	16.39	66.50%	59.50%	83.36
Multitasking	17.51	71.50%	57.30%	81.91
Structured Fusion (Frozen Modules)	17.53	65.80%	51.30%	76.08
Structured Fusion (Fine-tuned Modules)	18.51	77.30%	64.30%	89.31
Structured Fusion (Multitasked Modules)	16.70	80.40%	63.60%	88.71
Reinforcement Learning				
Seq2Seq + RL (Zhao et al., 2019)	1.40	80.50%	**79.07%**	81.19
LiteAttnCat + RL (Zhao et al., 2019)	12.80	**82.78%**	**79.20%**	**93.79**
Structured Fusion (Frozen Modules) + RL	**16.34**	**82.70%**	72.10%	**93.74**

Table 1: Experimental results for the various models. This table compares two classes of methods: those trained with supervised learning and those trained with reinforcement learning. All bold-face results are statistically significant ($p < 0.01$).

4.4 Results

Experimental results in Table 1 show that our Structured Fusion Networks (SFNs) obtain strong results when compared to both methods trained with and without the use of reinforcement learning. Compared to previous methods trained only with supervised learning, SFNs obtain a **+4.26** point improvement over seq2seq baselines in the combined score with strong improvement in both Success and Inform rates. SFNs are outperformed by the recently published HDSA (Chen et al., 2019) models which relies on BERT (Devlin et al., 2018) and conditioning on graph structured dialog acts. When using reinforcement learning, SFNs match the performance of LiteAttnCat (Zhao et al., 2019) on the combined score. Though the Inform rate is equivalent and the Success rate is lower (albeit still better than all supervised methods), the BLEU score of SFNs is much better with an improvement of **+3.54** BLEU over LiteAttnCat.

In the reinforcement learning setting, the improved BLEU can be attributed to the explicit structure of the model. This structure enables the model to optimize for the reward (Success rate) without resulting in degenerate output (Lewis et al., 2017).

SFNs obtain the highest combined score when the modules are fine-tuned. This is likely because, while the structured modules serve as a strong initialization for the task of dialog generation, forcing the model to maintain the exact structure (i.e., frozen modules) limits its ability to learn. In fact, the end-to-end model may choose to ignore some elements of intermediate structure (e.g., a particular dialog act) which prove useless for the task of response generation.

Despite strong overall performance, SFNs do show a **-2.27** BLEU drop when compared to the strongest seq2seq baseline and a **-5.09** BLEU drop compared to HDSA. Though it is difficult to ascertain the root cause of this drop, one potential reason could be that the dataset contains many social niceties and generic statements (e.g., *"happy anniversary"*) which are difficult for a structured model to effectively generate (since it is not an element of the structure) while a free-form sequence-to-sequence network would not have this issue.

To a lesser degree, multi-tasking (i.e., multitasked modules) would also prevent the model from being able to ignore some elements of the structure. However, the SFN with multitasked modules performs best on the Inform metric with a **+9.07%** improvement over the seq2seq baselines and a **+3.10%** over other SFN-based methods. This may be because the Inform metric measures how many of the requested attributes were answered, which benefits from a structured representation of the input.

Zero-Shot Naïve Fusion performs very poorly, suggesting that the individual components have difficulty producing good results when given imperfect input. Though the NLG module performs extremely well when given the oracle dialog acts (28.97 BLEU; 106.02 combined), its performance deteriorates significantly when given the predicted dialog acts. This observation is also applicable to Structured Fusion with frozen modules.

HDSA (Chen et al., 2019) outperforms SFN possibly due to the use of a more sophisticated Transformer model (Vaswani et al., 2017) and BERT pre-training (Devlin et al., 2018). A unique advantage of SFNs is that the architecture of the *neural dialog modules* is flexible. The performance of HDSA could potentially be integrated with SFNs by using the HDSA model as the NLG module of an SFN. This is left for future work, as the HDSA model was released while this paper was already in review.

These strong performance gains reaffirm the hypothesis that adding explicit structure to neural dialog systems results in improved modelling ability particularly with respect to dialog policy as we see in the increase in Inform and in Success. The results with reinforcement learning suggest that the explicit structure allows *controlled fine-tuning* of the models, which prevents divergent behavior and degenerate output.

4.5 Human Evaluation

To supplement the results in Table 1, human evaluation was used to compare seq2seq, SFN, SFN fine-tuned with reinforcement learning, and the ground-truth human response. Workers on Amazon Mechanical Turk (AMT) were asked to read the context, and score the *appropriateness* of each response on a Likert scale (1-5). One hundred context-response pairs were labeled by three workers each. The results shown in Table 2 demonstrate that SFNs with RL outperform the other methods in terms of human judgment. These results indicate that in addition to improving on automated metrics, SFNs result in user-favored responses.

Model	Avg Rating	≥ 4	≥ 5
Seq2Seq	3.00	40.21%	9.61%
SFN	3.02	**44.84%**	11.03%
SFN + RL	**3.12**	**44.84%**	**16.01%**
Human	3.76	59.79%	34.88%

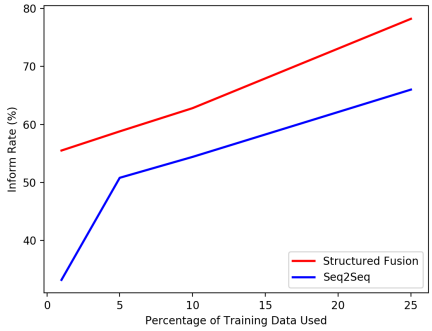
Table 2: Results of human evaluation experiments. The ≥ 4 and ≥ 5 columns indicate the percentage of system outputs which obtained a greater than 4 and 5 rating, respectively.

5 Analysis

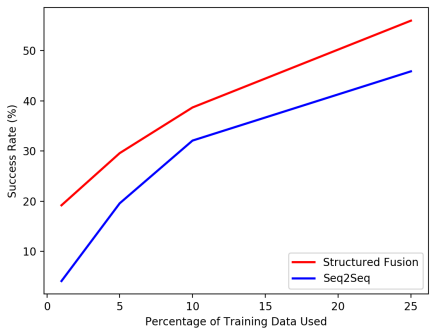
5.1 Limited Data

Structured Fusion Networks (SFNs) should outperform sequence-to-sequence (seq2seq) networks in reduced data scenarios due to the explicit structure. While a baseline method would require large amounts of data to learn to infer structure, SFNs do this explicitly.

The performance of seq2seq and SFNs are determined, when training on 1%, 5%, 10% and 25% of the training data (total size of $\sim$ 55,000 utterances). The supervised-learning variant of SFNs with fine-tuned modules is used. The pre-training of the modules and fine-tuning of the full model is done on the same data split. The full data is used during validation and testing.



(a)



(b)

Figure 6: Variation of Inform (a) and Success (b) rate at different amounts of training data.

The results in Figure 6 show the Inform and Success rates for different amounts of training data. SFNs significantly outperform the seq2seq model in low-data scenarios. Notably, improve-

ment is considerably higher in the most extreme low-data scenario, when only 1% of the training data ($\sim$ 550 dialogs) is used. As the amount of training data increases, the gap between the two models stabilizes. The effectiveness at extreme low-data scenarios reaffirms the hypothesis that explicit structure makes SFNs less data-hungry than sequence-to-sequence networks.

5.2 Domain Generalizability

The explicit structure of SFNs should facilitate effective domain generalizability. A domain transfer experiment was constructed to evaluate the comparative ability of seq2seq and SFNs. The models were both trained on a reduced dataset that largely consists of out-of-domain examples and evaluated on in-domain examples. Specifically, 2000 out-of-domain training examples and only 50 in-domain training examples were used. The restaurant domain of MultiWOZ was selected as in-domain.

Model	BLEU	Inform	Success
Seq2Seq	10.22	35.65%	1.30%
SFN	7.44	**47.17%**	**2.17%**

Table 3: Results of the domain transfer experiment comparing sequence-to-sequence and Structured Fusion Networks. All bold-face results are statistically significant ($p < 0.01$).

The results, seen on Table 3, show that SFNs perform significantly better on both the Inform (**+11.52%**) and Success rate. Although SFNs have a slightly higher Success rate, both models perform poorly. This is expected since the models would be unable to answer all the requested attributes when they have seen little domain data – their language model would not be tuned to the in-domain task. The **-2.78** BLEU reduction roughly matches the BLEU difference observed on the main task, therefore it is not an issue specific to domain transfer.

6 Conclusions and Future Work

This paper presents several methods of incorporating explicit structure into end-to-end neural models of dialog. We created Structured Fusion Networks, comprised of pre-trained *dialog modules* and a higher-level end-to-end network, which obtain strong results on the MultiWOZ dataset both with and without the use of reinforcement learning. SFNs are further shown to be robust to divergence during reinforcement learning, effective in low data scenarios and better than sequence-to-sequence on the task of domain transfer.

For future research, the explicit structure of SFNs has been shown to have multi-faceted benefits; another potential benefit may be interpretability. It would be interesting to investigate the use of SFNs as more interpretable models of dialog. While domain generalizability has been demonstrated, it would be useful to further explore the nature of generalizability (e.g., task transfer, language style transfer). Another potential avenue of research is whether the explicit structure of SFNs could potentially allow swapping the dialog modules without any fine-tuning. Structured Fusion Networks highlight the effectiveness of using explicit structure in end-to-end neural networks, suggesting that exploring alternate means of incorporating structure would be a promising direction for future work.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Dan Bohus and Alexander I Rudnicky. 2009. The ravenclaw dialog management framework: Architecture and systems. *Computer Speech & Language*, 23(3):332–361.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. *arXiv preprint arXiv:1905.12866*.

Jan Chorowski and Navdeep Jaitly. 2016. Towards better decoding and language model integration in sequence to sequence models. *arXiv preprint arXiv:1612.02695*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2019. The second conversational

intelligence challenge (convai2). *arXiv preprint arXiv:1902.00098*.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

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

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

Bing Liu and Ian Lane. 2016. Joint online spoken language understanding and language modeling with recurrent neural networks. *arXiv preprint arXiv:1609.01462*.

Shikib Mehri, Evgeniia Razumovsakaia, Tiancheng Zhao, and Maxine Eskenazi. 2019. Pretraining methods for dialog context representation learning. *arXiv preprint arXiv:1906.00414*.

Tom M Mitchell. 1980. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research .

Kaixiang Mo, Yu Zhang, Qiang Yang, and Pascale Fung. 2018. Cross-domain dialogue policy transfer via simultaneous speech-act and slot alignment. *arXiv preprint arXiv:1804.07691*.

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

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. Understanding the exploding gradient problem. *CoRR*.

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*, volume 16, pages 3776–3784.

Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. 2017. Cold fusion: Training seq2seq models together with language models. *arXiv preprint arXiv:1708.06426*.

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.

Jörg Tiedemann. 2009. News from opus-a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248.

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

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Zhuhao Wang, Fei Wu, Weiming Lu, Jun Xiao, Xi Li, Zitong Zhang, and Yueting Zhuang. 2016. Diverse image captioning via grouptalk. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2957–2964. AAAI Press.

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

Tiancheng Zhao and Maxine Eskenazi. 2018. Zero-shot dialog generation with cross-domain latent actions. *arXiv preprint arXiv:1805.04803*.

Tiancheng Zhao, Allen Lu, Kyusong Lee, and Maxine Eskenazi. 2017. Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability. *arXiv preprint arXiv:1706.08476*.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. *arXiv preprint arXiv:1902.08858*.

A Belief State Ablation Study

All previous research working on dialog generation for the MultiWOZ dataset uses the ground-truth belief state vector during training and evaluation. Therefore for fair comparability, the SFN experiments in our paper had the NLU module replaced by an "oracle NLU" which always outputs the ground-truth belief state.

An ablation experiment was performed to ascertain whether providing *only* the ground-truth belief state was the optimal solution. Several methods of combining the ground-truth belief state with the pre-trained NLU module were explored. These methods are enumerated as follows:

(1) **Ground-Truth Only:** The setting used in the primary experiments, shown in Table 1 of the main paper. Only the ground-truth belief state vector is used.

(2) **Predicted Only:** Only the belief state predicted by the pre-trained NLU module is used.

(3) **Sum:** The predicted and ground-truth belief states are summed, before being used by all upper layers.

(4) **Linear:** The predicted and ground-truth belief states area concatenated and passed through a linear layer.

These experiments are performed using the best model, Structured Fusion Networks with fine-tuned modules. The results are shown in Table 4.

Model	BLEU	Inform	Success	Comb.
GT	**18.51**	**77.30%**	**64.30%**	**89.31**
Pred	16.88	73.80%	58.60%	83.04
Sum	15.93	72.90%	60.80%	82.78
Linear	15.42	66.80%	54.80%	76.22

Table 4: Results of the domain transfer experiment comparing sequence-to-sequence and Structured Fusion Networks. All bold-face results are statistically significant ($p < 0.01$).

It is observed that adding the pre-trained NLU does not provide any additional performance benefit, when the ground-truth belief state is already provided. As such, combinations of the ground-truth and predicted belief state actually perform worse than either of the methods independently because of (1) additional parameters to be learned, especially in the case of the *Linear* method, and (2) a conflicting trade-off between fine-tuning a learned NLU module and using the ground-truth belief state.

B Qualitative Examples

Table 5 shows several examples of dialogs from the test set of MultiWOZ, along with the produced response from three different models: sequence-to-sequence networks, Structured Fusion Networks, and Structured Fusion Networks fine-tuned with reinforcement learning. These examples serve to provide insight into the respective strengths and weaknesses of the different models. A few noteworthy observations from the four examples are enumerated below:

(1) SFN fine-tuned with RL **consistently provides more attribute information.** It provides at least one attribute in every example response, for a total of 14 total attributes across the four examples. This, along with the high Success score of this model, is a consequence of the reinforcement learning fine-tuning which directly optimizes the Success score and rewards the model for providing the correct attributes.

(2) Seq2Seq **produces more generic responses**. In the second and fourth examples, the outputs produced by Seq2Seq are generic and are unrelated to the context. This suggests that the Seq2Seq model has an overwhelming implicit language model, which produces generic and dull responses. On the other hand, the explicit structure in SFN mitigates the effect of the implicit language model by relying on an *explicit* language model (in the form of the NLG).

(3) Seq2Seq **requests attributes which the user has already provided.** In the first and third example, the Seq2Seq output is requesting parameters that the user explicitly provided. This highlights the lack of structured representations in the Seq2Seq model. SFN which explicitly models the structure of the input effectively understands and captures the information provided in the context.

Example 1
Dialog Context: USER: hello . i am looking for a [value_pricerange] place to eat in the [value_area] . can you help me ?
Sequence-to-Sequence Response: i have [value_count] options for you . do you have a preference on price range ?
Structured Fusion Network Response: [restaurant_name] is a [value_pricerange] -ly priced [value_food] restaurant in the [value_area] . would you like to make a reservation ?
Structured Fusion Network w/ Reinforcement Learning Response: [restaurant_name] is a [value_pricerange] [value_food] restaurant in the [value_area] . it is located at [restaurant_address] . the postcode is [restaurant_postcode] . the phone number is [restaurant_phone]
Example 2
Dialog Context: . . . USER: i want to book this hotel for the same group of people and [value_count] nights starting from [value_day] . i'll need a reference number please . SYSTEM: your reservation is confirmed and the reference number is ## USER: i am sorry ? were you able to book the express by holiday inn for [value_count] people ?
Sequence-to-Sequence Response: you are welcome . have a great day !
Structured Fusion Network Response: you are all set . your reference number is [hotel_reference] . can i help you with anything else ?
Structured Fusion Network w/ Reinforcement Learning Response: your booking was successful . the reference number is [hotel_reference] .
Example 3
Dialog Context: USER: can you see if there are any entertainment attractions in the [value_area] of town ?
Sequence-to-Sequence Response: there are [value_count] museums in the [value_area] . do you have a preference for area ?
Structured Fusion Network Response: there are [value_count] museums in the [value_area] . [attraction_name] is a museum and the [attraction_name] is a [value_count] .
Structured Fusion Network w/ Reinforcement Learning Response: [attraction_name] is a museum in the [value_area] and the [attraction_name] is located at [attraction_address] . the postcode is [attraction_postcode] .
Example 4
Dialog Context: USER: i need a taxi . i'll be departing from [restaurant_name] . SYSTEM: i can help you with that . when do you need to leave ? USER: i would like to leave after [value_time] please . SYSTEM: where will you be going ? USER: i'll be going to [restaurant_name] .
Sequence-to-Sequence Response: you are welcome . have a good time !
Structured Fusion Network Response: you are welcome . have a great day !
Structured Fusion Network w/ Reinforcement Learning Response: i have booked for you a [taxi_type] , the contact number is [taxi_phone] . is there anything else i can help you with ?

Table 5: Four examples of dialog contexts from the dataset, and the responses generated by three different models: Seq2Seq, SFN, and SFN with RL.