A Copy-Augmented Sequence-to-Sequence Architecture Gives Good Performance on Task-Oriented Dialogue

Mihail Eric
Computer Science Department
Stanford University
meric@cs.stanford.edu

Christopher D. Manning
Computer Science Department
Stanford University
manning@stanford.edu

Abstract

Task-oriented dialogue focuses on conversational agents that participate in user-initiated dialogues on domain-specific topics. In contrast to chatbots, which simply seek to sustain open-ended meaningful discourse, existing task-oriented agents usually explicitly model user intent and belief states. This paper examines bypassing such an explicit representation by depending on a latent neural embedding of state and learning selective attention to dialogue history together with copying to incorporate relevant prior context. We complement recent work by showing the effectiveness of simple sequence-to-sequence neural architectures with a copy mechanism. Our model outperforms more complex memory-augmented models by 7% in per-response generation and is on par with the current state-of-the-art on DSTC2.

1 Introduction

Effective task-oriented dialogue systems are becoming important as society progresses toward using voice for interacting with devices and performing everyday tasks such as scheduling. To that end, research efforts have focused on using machine learning methods to train agents using dialogue corpora. One line of work has tackled the problem using partially observable Markov decision processes and reinforcement learning with carefully designed action spaces (Young et al., 2013). However, the large, hand-designed action and state spaces make this class of models brittle, and in practice most deployed dialogue system remain hand-written rule-based systems.

Recently, neural network models have achieved success on a variety of natural language processing tasks (Bahdanau et al., 2014; Sutskever et al., 2014; Vinyals et al., 2015b), due to their ability to implicitly learn powerful distributed representations from data in an end-to-end trainable fashion. This paper extends recent work examining the utility of distributed state representations for task-oriented dialogue agents, without providing rules or manually tuning features.

One prominent line of recent neural dialogue work has continued to build systems with modularly-connected representation, belief state, and generation components (Wen et al., 2016b). These models must learn to explicitly represent user intent through intermediate supervision, and hence suffer from not being truly end-to-end trainable. Other work stores dialogue context in a memory module and repeatedly queries and reasons about this context to select an adequate system response (Bordes and Weston, 2016). While reasoning over memory is appealing, these models simply choose among a set of utterances rather than generating text and also must have temporal dialogue features explicitly encoded.

We aim to fill a gap in the literature by systematically building increasingly complex models of text generation, starting with a vanilla sequence-to-sequence recurrent architecture. The result is a simple, intuitive, and highly competitive model, which outperforms the more complex model of Bordes and Weston (2016) by 6.9%. Our contributions are as follows: 1) We perform a systematic, empirical analysis of increasingly complex sequence-to-sequence models for task-oriented dialogue, and 2) we develop a recurrent neural dialogue architecture augmented with an attention-based copy mechanism that is able to significantly outperform more complex models on a variety of metrics on realistic data.

2 Architecture

We use neural encoder-decoder architectures to frame dialogue as a sequence-to-sequence learning problem. Given a dialogue between a user (*u*) and a system (*s*), we represent the dialogue utterances as $\{(u_1, s_1), (u_2, s_2), \ldots, (u_k, s_k)\}$ where k denotes the number of turns in the dialogue. At the i^{th} turn, we encode the aggregated dialogue context composed of the tokens of $(u_1, s_1, \ldots, s_{i-1}, u_i)$. Letting $x_1, \ldots, x_m$ denote these tokens, we first embed these tokens using a trained embedding function ϕ^{emb} that maps each token to a fixed-dimensional vector. These mappings are fed into the encoder to produce context-sensitive hidden representations $h_1, \ldots, h_m$.

The vanilla Seq2Seq decoder predict the tokens of the i^{th} system response s_i, which we denote as $y_1, \ldots, y_n$ with a recurrent model. We extend that with an attention-based model (Bahdanau et al., 2014; Luong et al., 2015a), where, at every time step j of the decoding, an attention score e_{ji} is computed for each hidden state h_i of the encoder, using the attention mechanism of (Vinyals et al., 2015b). These scores are used to form a normalized linear combination of the encoder hidden representations $h_1, \ldots, h_m$ which is linearly combined with the current hidden state $\tilde{h}_j$ of the decoder into a fixed-dimensional vector o_j. In the basic attention model, this o_j is used to compute logits over the tokens of the output vocabulary V, and the next token y_j is predicted by maximizing the log-likelihood.

An effective task-oriented dialogue system must have powerful language modelling capabilities and be able to pick up on relevant entities of an underlying knowledge base. We augment the attention encoder-decoder model with an attention-based copy mechanism in the style of (Jia and Liang, 2016). During decoding, we define $\tilde{o}_j = [o_j, e_j]$ where e_j is the combined attention scores of the encoder hidden states, and a new logits vector $d \in \mathbb{R}^{|V|+m}$ is computed as $U\tilde{o}_j$ where U is an appropriately-dimensioned trainable matrix. Thus the model either predicts a token y_j from V or softly copies a token x_i from the encoder input context, via the attention score e_{ji}. Rather than copy over any token mentioned in the encoder dialogue context, our model is trained to only copy over entities of the knowledge base mentioned, as this provides a conceptually intuitive goal for the model's predictive learning: as training progresses it will learn to either predict a token from the standard vocabulary of the language model thereby ensuring well-formed natural language utterances, or to copy over the relevant entities from the input context, thereby learning to extract important dialogue context.

In our best performing model, we augment the inputs to the encoder by adding entity type features. Classes present in the knowledge base of the dataset are encoded as one-hot vectors. Whenever an entity token is seen during encoding, we append the appropriate one-hot vector to the token's word embedding before it is fed into the recurrent cell.

All of our architectures use an LSTM cell as the recurrent unit (Hochreiter and Schmidhuber, 1997) with a bias of 1 added to the forget gate in the style of (Zaremba et al., 2015).

2.1 Training

We train using a cross-entropy loss and the Adam optimizer (Kingma and Ba, 2015), applying dropout (Hinton et al., 2012) as a regularizer to the input and output of the LSTM. We identified hyperparameters by random search, evaluating on a held-out validation subset of the data. Dropout keep rates ranged from 0.75 to 0.95. We used word embeddings with size 300, and hidden layer and cell sizes were set to 353, identified through our search. We applied gradient clipping with a clip-value of 10 to avoid gradient explosions during training. The attention, output parameters, word embeddings, and LSTM weights are randomly initialized from a uniform unit-scaled distribution in the style of (Sussillo and Abbott, 2015).

3 Experiments

3.1 Data

For our experiments, we used dialogues extracted from the Dialogue State Tracking Challenge 2 (DSTC2) (Henderson et al., 2014), a restaurant reservation system dataset. While the goal of the original challenge was building a system for inferring dialogue state, for our study, we use the version of the data from Bordes and Weston (2016), which ignores the dialogue state annotations, using only the raw text of the dialogues, while adding system commands. Thus, the raw text includes user and system utterances as well as the API calls the system would make to the underlying KB in response to the user's queries. We use the train/validation/test splits from this mod-

Avg. # of Utterances Per Dialogue	14
Vocabulary Size	1,229
Training Dialogues	1,618
Validation Dialogues	500
Test Dialogues	1,117
# of Distinct Entities	452
# of Entity (or Slot) Types	8

Table 1: Statistics of DSTC2

ified version of the dataset. The dataset is appealing for a number of reasons: 1) It is derived from a real-world system so it presents the kind of linguistic diversity and conversational abilities we would hope for in an effective dialogue agent. 2) It is grounded via an underlying knowledge base of restaurant entities and their attributes. 3) Previous results have been reported on it so we can directly compare our model performance. We include statistics of the dataset in Table 1.

3.2 Metrics

Evaluation of dialogue systems is known to be difficult (Liu et al., 2016). We employ several metrics for assessing specific aspects of our model, drawn from previous work:

- **Per-Response Accuracy**: Bordes and Weston (2016) report a per-turn response accuracy, which tests their model's ability to select the system response at a certain timestep. Their system does a multiclass classification over a predefined candidate set of responses, which was created by aggregating all system responses seen in the training, validation, and test sets. Our model actually generates each individual token of the response, and we consider a prediction to be correct if every token of the model output matches the corresponding token in the gold response. Evaluating using this metric on our model is therefore significantly more stringent.

- **Per-Dialogue Accuracy**: Bordes and Weston (2016) also report a per-dialogue accuracy, which assesses their model's ability to classify every system turn of the dialogue correctly. We calculate a similar value of dialogue accuracy, though again our model generates every token of every response.

- **BLEU**: We use the BLEU metric, commonly employed in evaluating machine translation systems (Papineni et al., 2002), which has also been used in past literature for evaluating dialogue systems (Ritter et al., 2011; Li et al., 2015). We calculate average BLEU score over all responses generated by the system, and primarily report these scores to gauge our model's ability to accurately generate the language patterns seen in DSTC2.

- **Entity $\mathbf{F}_1$**: We also report entity F_1 averaged over all responses, to evaluate the model's ability to generate relevant entities from the underlying knowledge base and to capture the semantics of the user-initiated dialogue flow.

Our experiments show that sometimes our model generates a response to a given input that is perfectly reasonable, but is penalized because our evaluation metrics involve direct comparison to the gold system output. For example, given a user request for an *australian restaurant*, the gold system output is *you are looking for an australian restaurant right?* whereas our system outputs *what part of town do you have in mind?*, which is a more directed follow-up intended to narrow down the search space of candidate restaurants the system should propose. This issue, which recurs with evaluation of dialogue or other generative systems, could be alleviated through more forgiving evaluation procedures based on beam search decoding.

3.3 Results

In Table 2, we include the results of our models compared to the reported performance of the best performing model of (Bordes and Weston, 2016), which is a variant of an end-to-end memory network (Sukhbaatar et al., 2015). Their model is referred to as *MemNN*. We also include the model of (Liu and Perez, 2016), referred to as *GMemNN*, and the model of (Seo et al., 2016), referred to as *QRN*, which currently is state-of-the-art. In the table, Seq2Seq refers to our vanilla encoder-decoder architecture with (1), (2), and (3) LSTM layers respectively. +Attn refers to a 1-layer Seq2Seq with attention-based decoding. +Copy refers to +Attn with our copy-mechanism added. +EntType refers to +Copy with entity class features added to encoder inputs.

We see that a 1-layer vanilla encoder-decoder is already able to significantly outperform *MemNN* in both per-response and per-dialogue accuracies, despite our more stringent setting. Adding layers to Seq2Seq leads to a drop in performance, suggesting an overly powerful model for the small

Data	Model	Per-Resp.	Per Dial.	BLEU	Ent. F_1
Test set	*MemNN*	41.1	0.0	–	–
	GMemNN	48.7	1.4	–	–
	QRN	50.7	–	–	–
	Seq2Seq (1)	46.4	1.5	55.0	69.7
	Seq2Seq (2)	43.5	1.3	54.2	67.3
	Seq2Seq (3)	44.2	**1.7**	55.4	65.9
	+ Attn.	46.0	1.4	**56.6**	67.1
	+ Copy	47.3	1.3	55.4	71.6
	+ EntType	**48.0**	1.5	56.0	**72.9**
Dev set	Seq2Seq (1)	57.0	3.6	72.1	68.7
	Seq2Seq (2)	54.1	3.0	71.3	66.3
	Seq2Seq (3)	54.0	3.2	71.5	64.3
	+ Attn.	55.2	3.4	71.9	66.1
	+ Copy	58.9	3.6	73.1	72.5
	+ EntType	59.2	3.4	72.7	72.3

Table 2: Evaluation on DSTC2 test (top) and dev (bottom) data. Bold values indicate our best performance. A dash indicates unavailable values.

dataset size. Adding an attention-based decoding to the vanilla model increases BLEU although per-response and per-dialogue accuracies suffer a bit. Adding our attention-based entity copy mechanism achieves large increases in per-response accuracies and entity F_1. Adding entity class features to +Copy achieves our best-performing model, in terms of per-response accuracy and entity F_1. This model achieves a 6.9% increase in per-response accuracy on DSTC2 over *MemNN*, including +1.5% per-dialogue accuracy, and is on par with the performance of *GMemNN*, including beating its per-dialogue accuracy. It also achieves the highest entity F_1.

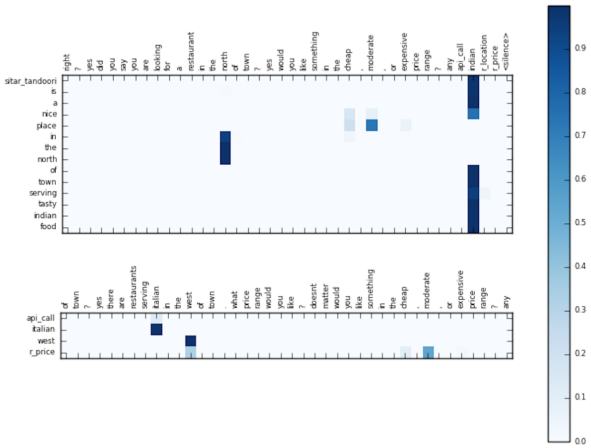
4 Discussion and Conclusion

We have iteratively built out a class of neural models for task-oriented dialogue that is able to outperform other more intricately designed neural architectures on a number of metrics. The model incorporates in a simple way abilities that we believe are essential to building good task-oriented dialogue agents, namely maintaining dialogue state and being able to extract and use relevant entities in its responses, without requiring intermediate supervision of dialogue state or belief tracker modules. We attribute the large gains in per-response accuracy and entity F_1 demonstrated by +EntType to its ability to pick out the relevant KB entities from the dialogue context fed into the encoder. In Figure 1, we see the attention-based copy weights of the model, indicating that the model is able to learn the relevant entities it should focus on in the input context. The powerful language modelling

cheap restaurant in east part of town
api_call r_cuisine east cheap
<silence>
the_missing_sock is a nice place in the east of town and the prices are cheap
address
sure, the_missing_sock is on the_missing_sock_address
phone number
the phone number of the_missing_sock is the_missing_sock_phone
thank you good bye
you are welcome

Table 3: Sample dialogue generated. System responses are in italics. The dataset uses fake addresses and phone numbers.

Figure 1: Attention-copy weights for a generated natural language response (top) and API call (bottom). The decoder output is displayed vertically and the encoder input is abbreviated for display.



abilities of the Seq2Seq backbone allow smooth integration of these extracted entities into both system-generated API calls and natural language responses as shown in the figure.

The appeal of our model comes from the simplicity and effectiveness of framing system response generation as a sequence-to-sequence mapping with a soft copy mechanism over relevant context. Unlike the task-oriented dialogue agents of Wen et. al (2016b), our architecture does not explicitly model belief states or KB slot-value trackers, and we preserve full end-to-end-trainability. Further, in contrast to other referenced work on DSTC2, our model offers more linguistic versatility due to its generative nature while still remaining highly competitive and outperforming other models. We hope this simple and effective architecture can be a strong baseline for future research efforts on task-oriented dialogue.

References

[Bahdanau et al.2014] D. Bahdanau, K. Cho, and Y. Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[Bordes and Weston2016] A. Bordes and J. Weston. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.

[Henderson et al.2014] M. Henderson, B. Thomson, and J. Williams. 2014. The second dialog state tracking challenge. *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 263.

[Hinton et al.2012] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

[Hochreiter and Schmidhuber1997] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, pages 1735–1780.

[Jia and Liang2016] R. Jia and P. Liang. 2016. Data recombination for neural semantic parsing. *Association for Computational Linguistics*.

[Kingma and Ba2015] D. Kingma and J. Ba. 2015. Adam: a method for stochastic optimization. *International Conference for Learning Representations*.

[Li et al.2015] J. Li, M. Galley, C. Brockett, J. Gao, and W. B. Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

[Liu and Perez2016] F. Liu and J. Perez. 2016. Gates end-to-end memory networks. *arXiv preprint arXiv:1610.04211*.

[Liu et al.2016] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau. 2016. How not to evaluate your dialogue system: an empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.

[Luong et al.2015a] M. Luong, H. Pham, and C.D. Manning. 2015a. Effective approaches to attention-based neural machine translation. *Empirical Methods in Natural Language Processing*, pages 1412–1421.

[Papineni et al.2002] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. *Association for Computational Linguistics*.

[Ritter et al.2011] A. Ritter, C. Cherry, and W. B. Dolan. 2011. Data-driven response generation in social media. *Empirical Methods in Natural Language Processing*.

[Seo et al.2016] M. Seo, S. Min, A. Farhadi, and H. Hajishirzi. 2016. Query-reduction networks for question answering. *arXiv preprint arXiv:1606.04582*.

[Sukhbaatar et al.2015] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. 2015. End-to-end memory networks. *arXiv preprint arXiv:1503.08895*.

[Sussillo and Abbott2015] D. Sussillo and L.F. Abbott. 2015. Random walk initialization for training very deep feed forward networks. *arXiv preprint arXiv:1412.6558*.

[Sutskever et al.2014] I. Sutskever, O. Vinyals, and Q.V. Le. 2014. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, pages 3104–3112.

[Vinyals et al.2015b] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton. 2015b. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2755–2763.

[Wen et al.2016b] T.H. Wen, M. Gasic, N. Mrksic, L. M. R.-B., P.-H. Su, S. Ultes, D. Vandyke, and S. Young. 2016b. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.

[Young et al.2013] S. Young, M. Gasic, B. Thomson, and J.D. Williams. 2013. POMDP-based statistical spoken dialog systems: a review. *Proceedings of the IEEE*, 28(1):114–133.

[Zaremba et al.2015] W. Zaremba, I. Sutskever, and O. Vinyals. 2015. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.