A Sequential Neural Approach on Task-Oriented Dialogue

Frank T. Tillman,

Cardinal Gibbons High School, North Charleston, South Carolina

ABSTRACT

Task-oriented dialogue focuses on conversational agents that participate in dialogues with user goals on domain-specific topics. In reference 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, a real world task-oriented dialogue dataset

1. PREFACE

Effective task-oriented dialogue systems are becoming important as society progresses toward us-ing voice for interacting with devices and perform-ing 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 prob-lem using partially observable Markov decision processes and reinforcement learning with carefully designed action spaces.

However, the large, hand-designed action and state spaces make this class of models brittle and are not scalable, and in practice most deployed dialogue systems remain handwritten, rule based systems. Recently, neural network models have achieved success on a variety of natural language processing tasks, due to their ability to implicitly learn powerful distributed representations from data in an end-to-end trainabe 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. These models must learn to explicitly represent user intent through intermediate supervision, and hence suffer from not being truly end-to-end train-able. Other work stores dialogue context in a memory module and repeatedly queries and reasons about this context to select an adequate system response. 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.

However, the present literature lacks results for now standard sequence-to-sequence architectures, and we aim to fill this gap by building increasingly complex models of text generation, starting with a vanilla sequence-to-sequence recurrent architec-ture. 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.
2. We develop a recur-rent 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 $\{(U_1, S_1), (U_2, S_2), \dots ,(U_k, S_k)\}$ where k denotes the number of turns in the dialogue. At the i^{th} turn of the dialogue, we encode the aggregated dialogue context composed of the tokens of (U_1,S_1,…,$S_{i−1}$, U_i). Letting x_1, …, 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,…, h_m.

The vanilla Seq2Seq decoder predicts the tokens of the i^{th} system response S_i by first computing decoder hidden states via the recurrent unit. We denote h_1,…, h_n as the hidden states of the decoder and y_1, … , y_n as the output tokens. We extend this decoder with an attention- based model, where, at every time step t of the decoding, an attention score a^t_i is computed for each hidden state h_i of the encoder, using the attention mechanism of (Vinyals et al., 2015). Formally this attention can be described by the following equations:

$$u_i^t = v^T \tanh(W_1 h_i + W_2 \tilde{h}_t) \qquad (1)$$

$$a_i^t = \text{Softmax}(u_i^t) \qquad (2)$$

$$\tilde{h}'_t = \sum_{i=1}^{m} a_i^t h_i \qquad (3)$$

$$o_t = U[\tilde{h}_t, \tilde{h}'_t] \qquad (4)$$

$$y_t = \text{Softmax}(o_t) \qquad (5)$$

Where W_1, W_2, U, and v are trainable parameters of the model and o_t represents the logits over the tokens of the output vocabulary V. During training, the next token y_t is predicted so as to maximize the log likelihood of the correct output sequence given the input sequence. 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.

One source of relevant entities is that they will commonly have been mentioned in the prior discourse context.

Recent literature has shown that incorporating a copying mechanism into neural architectures improves performance on various sequence-to-sequence tasks including code generation, machine translation, and text summarization. We therefore augment the attention encoder-decoder model with an attention-based copy mechanism in the style of (Jia and Liang, 2016). In this scheme, during decoding we compute our new logits vector as

$$o_t = U[\tilde{h}_t, \tilde{h}'_t, a^t_{[1:m]}]$$

Where $a^t_{[1:m]}$ is the concatenated attention scores of the encoder hidden states, and we are now predicting over a vocabulary of size |V | + m. The model, thus, either predicts a token y_t from V or copies a token x_i from the encoder input context, via the attention score a^t_i . 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 in the dialogue context, 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, namely the 8 distinct entity types referred to in Table 1, are encoded as one hot vectors. Whenever a token of a certain entity type 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. These type features improve generalization to novel entities by allowing the model to hone in on positions with particularly relevant bits of dialogue context during its soft attention and copying. Other cited work using the DSTC2 dataset implement similar mechanisms whereby they expand the feature representations of candidate system responses based on whether there is lexical entity class matching with provided dialogue context

3. RESEARCH AND ANALYSIS.

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. 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. Our model then aims to predict both these system utterances and API calls, each of which is regarded as a turn of the dialogue. We use the train/validation/test splits from this modified 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.

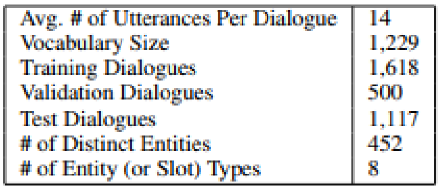
3. 3. Previous results have been reported on it so we can directly compare our model performance. We include statistics of the dataset in Table 1.

3.2 Learning And Development

We trained 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 hyper parameters 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 clipvalue of 10 to avoid gradient explosions during training. The attention, output parameters, word embeddings, and LSTM weights were randomly initialized from a uniform unit-scaled distribution in the style of (Sussillo and Abbott, 2015).

3.3 Model 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:



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

- **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 only 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 a test than for the model of Bordes and Weston (2016).

- **Per-Dialogue Accuracy:** Bordes and Weston (2016) also report a per-dialogue accuracy, which assesses their model's ability to produce every system response 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., 2016). 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 F1:** Each system response in the test data defines a gold set of entities. To compute an entity F1, we micro-average over the entire set of system dialogue responses. This metric evaluates the model's ability to generate relevant entities from the underlying knowledge base and to capture the semantics of the user-initiated dialogue flow.

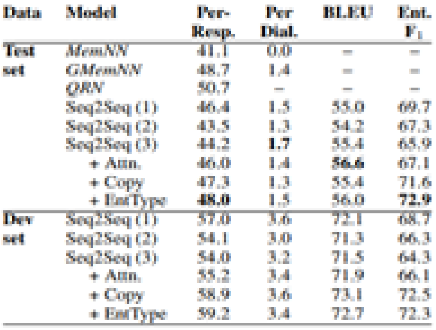
Our experiments show that sometimes our model generates a response to a given input that is perfectly reasonable, but is penalized because our evaluation metrics involve direct comparison to the gold system output. For example, given a user request for an australian restaurant, the gold system output 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.

3.4 Reports

In Table 2, we present 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 the stateof-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 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 substantial increases in perresponse accuracies and entity F1.



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. including beating its per-dialogue accuracy. It also achieves the highest entity F1.

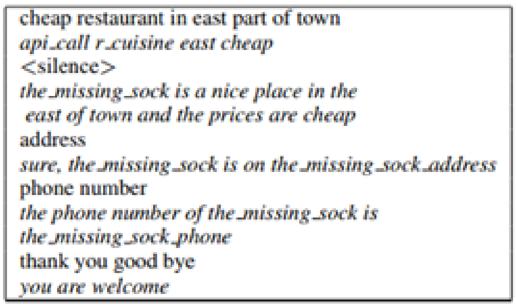
4 CONCLUSION

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.

Other dialogue models tested on DSTC2 that are more performant in per-response accuracy are equipped with sufficiently more complex mechanisms than our model. GMemNN uses an explicit memory module as well as an adaptive gating mechanism to learn to attend to relevant memories.

We contrast with these works by bootstrapping off of more empirically accepted Seq2Seq architectures through intuitive extensions, while still producing highly competitive models. We attribute the large gains in per response accuracy and entity F1 demonstrated by our +EntType to its ability to pick out the relevant KB entities from the dialogue context fed into the encoder.



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.

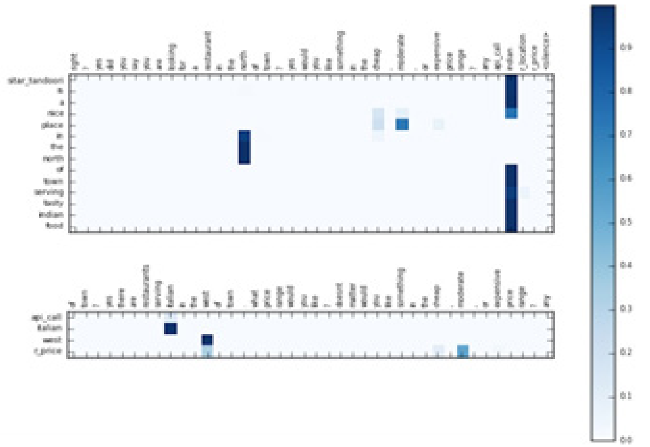


Figure 1: Attention-copy weights for a generated natural language response (top) and API call (bottom). The decoder output is displayed vertically and the encoder input is abbreviated for display weights of the model, indicating that the model is able to learn the relevant entities it should focus on in the input context.

The powerful language modelling 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. Of course, this is not to deny the importance of dialogue agents which can more effectively use a knowledge base to answer user requests, and this remains a good avenue for further work. Nevertheless, we hope this simple and effective architecture can be a strong baseline for future research efforts on task-oriented dialogue.

ACKNOWLEDGMENTS

I would like to thank Mihail Eric, Christopher D. Manning, Alex, Lakshmi Krishnan, Francois Charette, for their effort in the research in this field and their great contribution towards Data Science.

REFERENCES

1. Dzimitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In International Conference on Learning Representations (ICLR2015).

2. Antoine Bordes and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. arXiv preprint arXiv:1605.07683.

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

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

5. Matthew Henderson, Blaise Thomson, and Jason Williams. 2014. The second dialog state tracking challenge. 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue, page 263.

6. Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580. Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short- term memory. Neural Computation, pages 1735–1780.

7. Mihail Eric and Christopher D. Manning. A Copy-Augmented Sequence-to-Sequence Architecture Gives Good Performance on Task-Oriented Dialogue. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2122–2132, Austin, Texas, November. Association for Computational Linguistics.

8. Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Effective approaches to attentionbased neural machine translation. Empirical Methods in Natural Language Processing, pages 1412– 1421.

9. Vu Pham, Theodore Bluche, Christopher Kermorvant, and Jerome Louradour. 2014. Dropout improves recurrent neural networks for handwriting recognition. arXiv preprint arXiv:1312.4569v2.

10. Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media.

11. Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 27, pages 3104–3112. Curran Associates, Inc.

12. Oriol Vinyals, Ł ukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R.

13. Garnett, editors, Advances in Neural Information Processing Systems 28, pages 2773–2781. Curran Associates, Inc.

14. Tsung-Hsien Wen, David Vandyke, Milica Gasic, Nikola Mrksic, Lina. M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016b. A network-based end-to-end trainable task-oriented dialogue system. arXiv preprint arXiv:1604.04562.