What we need to learn if we want to *do* and not just *talk*

Rashmi Gangadharaiah, Balakrishnan (Murali) Narayanaswamy, Charles Elkan
Amazon AI Lab
{rgangad@,muralibn@,elkanc@}amazon.com

Abstract

In task-oriented dialog, agents need to generate both fluent natural language responses and correct external actions like database queries and updates. We show that methods that achieve state of the art performance on synthetic datasets, perform poorly in real world dialog tasks. We propose a hybrid model, where nearest neighbor is used to generate fluent responses and Sequence-to-Sequence (Seq2Seq) type models ensure dialogue coherency and generate accurate external actions. The hybrid model on an internal customer support dataset achieves a 78% relative improvement in fluency, and a 200% improvement in external call accuracy.

1 Introduction

Many commercial applications of artificial agents require task-oriented conversational agents that help customers achieve a specific goal, such as making or cancelling a payment or reservation (Zue et al., 2000; Bennacef et al., 1996). These chatbots must extract relevant information from the user, provide relevant knowledge to her, and issue appropriate system calls to achieve the goal.

Supervised approaches such as seq2seq models (Vinyals and Le, 2015; Shang et al., 2015; Serban et al., 2015; Sordoni et al., 2015b), have recently gained attention in non-task oriented dialog, due to their ability to perform end-to-end learning from expert dialogues[1], removing the need for many of the independent modules in traditional systems such as, natural language understanding, dialog state tracker and natural language generator.

Seq2Seq models have also shown promising results on small domain or synthetic task-oriented dialog datasets. However, performance was much worse when we applied these models to real world datasets. This is in part because end-to-end methods, in general, require large amounts of data before they are able to generate fluent textual responses. In real world settings, words chosen by human users and agents are not constrained to a fixed vocabulary, and hence we see many lexical variations even among semantically similar dialogs.

To ensure that information is both conveyed and understood, we want responses to be *fluent* as well as *coherent*. We say a response is *coherent* if it is a sensible response in the dialogue context. Table 1 shows responses generated by a variant of the seq2seq model, when trained on real customer-agent chat transcripts. The response of the chatbot during the fourth turn[2] in Table 1, accepting the customer's expression of gratitude, is coherent and fluent. Coherence of a response does not necessarily guarantee fluency. The generated response during the second turn is coherent but not fluent.

On our customer support dataset, seq2seq models performed well with salutations, but performed poorly both in terms of fluency and coherency on intermediate responses. The reason being, salutations contain minimal lexical variations across dialogs and occur more frequently when compared to other utterances. (Koehn and Knowles, 2017) use beam search decoding in Neural Machine Translation to mitigate fluency issues on larger translation datasets. Typically increasing the beam size improves translation quality, however, increasing beam sizes in Neural MT has shown to produce poor translations (Koehn and Knowles, 2017).

We propose nearest neighbor based approaches that can directly use and replay available expert utterances. This removes the need for the models to learn the grammar of the language, and allows the models to focus on learning what to say, rather than how to say it. The nearest neighbor-based

[1] We refer to an entire session of text exchanges between an agent and a customer as a *dialogue*.

[2] We define a *turn* as a pair of text exchanges between the customer and the agent.

Proceedings of NAACL-HLT 2018, pages 25–32
New Orleans, Louisiana, June 1 - 6, 2018.

Table 1: Generated dialogue on customer support data. *masked* used for anonymization.

Turn	User's Utterance	Agent's Response	Generated Chatbot Response
1	got charged for <masked> membership after trial i did not want to continue.	hello , my name is <PERSON> . i 'm here to help you today .	hello <PERSON> , my name is <PERSON> . i 'm here to help you .
2	<SILENCE>	thank you for being a <masked> member .	thank , for you <masked> member
3	<SILENCE>	i 'm sorry to hear that you were charged with our membership . no worries . i 'll do my best	<SILENCE>
4	thank you , i did not know that it would auto renew after the trial	you 're welcome	you 're most welcome
5	<SILENCE>	*api_call* cancel_refund	*api_call* cancel_refund
6	<SILENCE>	i hope my actions helped you out today	you 're most welcome
7	very appreciated	thank you for contacting <masked>	thank you for contacting <masked>

methods we propose naturally generate more fluent responses, since they use actual agent responses. However, our results in Table 3 show that they perform poorly in predicting external actions and at ensuring dialogue level coherency. In contrast, the skip-connection seq2seq models we propose here, learn when to produce external actions and produce more coherent dialogues. We propose a hybrid model that brings together the strengths of both the approaches.

The contributions of this paper are as follows:

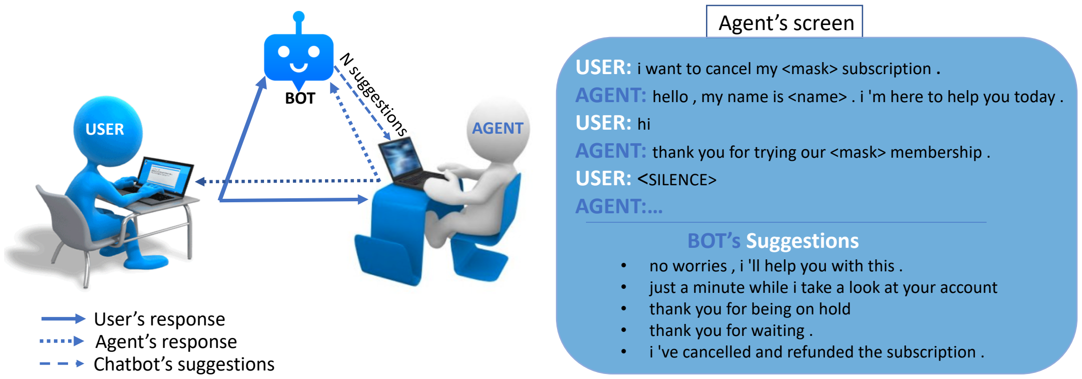
- We propose skip-connections to handle multi-turn dialogue that outperforms previous models.
- We propose a hybrid model where nearest neighbor-based models generate fluent responses and skip-connection models generate accurate responses and external actions. We show the effectiveness of the belief state representations obtained from the skip-connection model by comparing against previous approaches.
- To the best of our knowledge, our paper makes the first attempt at evaluating state of the art models on a large real world task with human users. We show that methods that achieve state of the art performance on synthetic datasets, perform poorly in real world dialog tasks. Comparing Tables 2 and 3, we see the impact of moving from synthetic to real world datasets, and as a result, find issues with previously proposed models that may have been obscured by the simplicity and regularity of synthetic datasets.

2 Related Work

Although seq2seq models have been applied in task-oriented settings (Wen et al., 2017; Williams and Zweig, 2016; Bordes and Weston, 2016; Zhao and Eskénazi, 2016), they have only been evaluated on small domain or synthetic datasets.

More recent work has focused on representation learning for multi-turn dialogue. Sordoni et al. (2015b) use a single bag-of-words representation of the entire dialog history. Such a representation ignores the order of responses, which is crucial to ensure that utterances are coherent across turns. An alternative approach is to use a hierarchical encoder-decoder network (HRED) (Sordoni et al., 2015a) which uses a complex three layered RNN network, a query level encoder, a session level encoder and a decoder. Attentional networks (Bordes and Weston, 2016; Dodge et al., 2015) use a weighted combination of all the context vectors upto the current turn. Attentional networks proved to be a stronger baseline over HRED during our evaluation. We propose models that learn fixed size representations of the history using simpler skip-connection models showing comparable performance with attentional networks (Bordes and Weston, 2016; Dodge et al., 2015).

Our work is closely related to retrieval-based chatbots. Williams and Zweig (2016), select a response from a small set of templates. Zhou et al. (2016); Yan et al. (2016) perform multi-turn dialogue by treating the dialogue history as the query, and perform classification with the number of classes equal to the number of possible responses. They evaluate precision@K, from a restricted list, but do not indicate how this list is obtained in practice. In our real world dataset, the number of possible responses grows with the dataset size. In addition, responses are unevenly distributed with salutations occurring frequently. As a



USER: i want to cancel my <mask> subscription .
AGENT: hello , my name is <name> . i 'm here to help you today .
USER: hi
AGENT: thank you for trying our <mask> membership .
USER: <SILENCE>
AGENT:...

BOT's Suggestions

- no worries , i 'll help you with this .
- just a minute while i take a look at your account
- thank you for being on hold
- thank you for waiting .
- i 've cancelled and refunded the subscription .

Figure 1: System Description. A human agent plays an intermediary role between the chatbot and the user.

result, the classification based approach performed poorly, with most of the outputs being salutations.

3 Proposed Approach

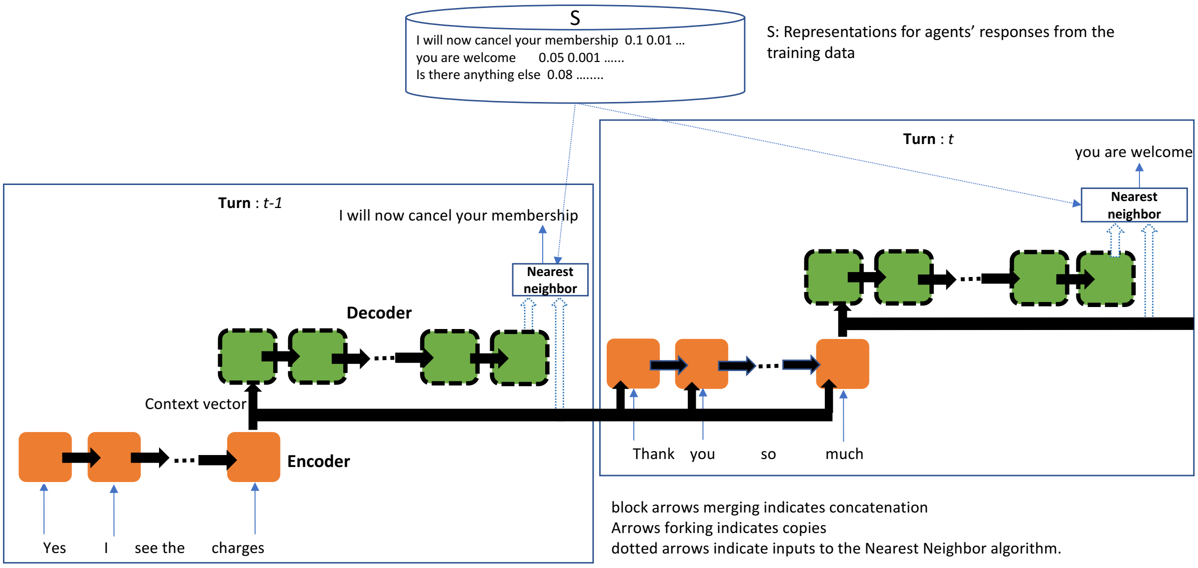
Complete automation of customer service is still not possible as chatbots are not perfect yet. However, automation where possible in the workflow could still result in considerable savings. In order to ensure that the end user experience is not substandard, in live user testing, we ask a human agent to play intermediary role between the chatbot and the user. A user initiates a chat by entering an initial query or an issue that requires resolution (Figure 1). The chatbot responds with 5 diverse responses. The agent selects the most relevant response, and may choose to modify it. If the response is not relevant, she may type a different response. During offline testing, the chatbot returns only one response and no human agent is used. The following section describes our skip connection seq2seq model for representation learning and our nearest neighbor approach for response selection. First we describe the datasets and metrics we use.

3.1 Dataset and Metrics

We use data from bAbI (Task1 and Task2) (Bordes and Weston, 2016) to evaluate our models. Other dialog tasks in bAbI require the model to mimic a knowledge base i.e., memorize it. This is not a suitable strategy for our application, since in practice knowledge bases undergo frequent changes, making this infeasible. In the bAbI task, the user interacts with an agent in a simulated restaurant reservation application, by providing her constraints, such as place, cuisine, number of people or price range. The agent or chatbot performs external actions or SQL-like queries (api_call) to retrieve information from the knowledge base of restaurants. We used 80% of the data for training (of which 10% was used for validation) and the remaining 20% for testing.

We also evaluate our models on an internal customer support dataset of 160k chat transcripts containing 3 million interactions. We limit the number of turns to 20. We will refer to this dataset as CS_large. We perform spell correction, deidentification to remove customer sensitive information, lexical normalization particularly of lingo words such as, *lol* and *ty*. Generalizing such entities reduces the amount of training data required. The values must be reinserted, currently by a human in the loop. We have also masked product and the organization name in the examples.

The use of MT evaluation metrics to evaluate dialogue fluency with just one reference has been debated (Liu et al., 2016). There is still no good alternative to evaluate dialog systems, and so we continue to report fluency using BLEU (BiLingual Evaluation Understudy (Papineni et al., 2002)), in addition to other metrics and human evaluations. Coherency also requires measuring correctness of the external actions which we measure using a metric we call, Exact Query Match (EQM), which represents the fraction of times the api_call matched the ground truth query issued by the human agent. We do not assign any credit to partial matches. In addition, we report the precision (P), recall (R) and accuracy (Acc) achieved by the models in predicting whether to make an api_call (positive) or not (negative). Obtaining and aligning api_calls with the chat transcripts is often complex as such information is typically stored in multiple confidential logs. In order to measure coherency with respect to api_calls, we randomly sampled 1000 chat tran-


S
I will now cancel your membership 0.1 0.01 ...
you are welcome 0.05 0.001 ......
Is there anything else 0.08 ....
Turn : t-1
I will now cancel your membership
Nearest neighbor
Decoder
Context vector
Encoder
Yes
I
see the
charges
Turn : t
you are welcome
Nearest neighbor
Thank
you
so
much

S: Representations for agents' responses from the training data

block arrows merging indicates concatenation
Arrows forking indicates copies
dotted arrows indicate inputs to the Nearest Neighbor algorithm.

Figure 2: Proposed embeddings for finding the nearest neighbor.

scripts and asked human agents to hand annotate the api_calls wherever appropriate. We will refer to this labeled dataset as CS_small.

3.1.1 Skip Connection Seq2Seq Model

Seq2seq models are an application of Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) architecture where inputs and outputs are variable length sequences. We unroll the basic seq2seq model and make one copy for each turn. This is illustrated in Figure 2. Input words are one hot encoded, and projected using a linear layer to obtain x_k^t for the input word at position k in turn t, resulting in a sequence $X_t = \{x_1^t, x_2^t, ...x_L^t\}$. The output sequence to be generated is represented by $Y_t = \{y_1^t, y_2^t, ...y_{L'}^t\}$. The encoder at turn t receives the user's projected input, as well as the context vectors from the final hidden units of the encoder and the decoder at turn $t-1$, forming a skip connection. This ensures that a fixed size vector is used to represent the dialogue history at every turn. Orange-solid-square boxes in Figure 2 represent LSTM cells of the encoder. $h_{L,enc}^t$ is the context vector which is sent to every LSTM cell in the decoder (dec) at any turn t (Cho et al., 2014).

Green-dashed-square cells in the decoder represent the LSTM and dense layers with a softmax non-linearity. These are trained to predict each word in the agent's utterance. Each of the seq2seq copies share the same parameters. Once the training is complete, we use only one copy of the seq2seq model to make predictions.

3.1.2 Results with Skip-Connections

The results obtained with the vanilla seq2seq model on the bAbI dataset is shown in the first row (**Model 1**) of Table 2. The EQM is 0%, even though the BLEU scores look reasonable. **Model 2** is the skip-connection seq2seq model, where only the output of the hidden states from the decoder at turn $t-1$ is appended to the input at time t, i.e., $h_{L,enc}^{t-1}$ from the *encoder* history is not explicitly presented to turn t.

Model 3 extends Model 1 by adding an attentional layer. Model 3 is a variant of Bordes and Weston (2016); Dodge et al. (2015) where the output of the attentional layer is sent to the decoder for generating the responses rather than classifying as one of the known responses. This variant performed better on the customer support data compared to a direct implementation of Bordes and Weston (2016). The reason being, salutations occurred more frequently in the customer support data and hence, the classification based approach originally proposed by Bordes and Weston (2016) classified most of the outputs as salutations. Finally, **Model 4** extends Model 2 by providing $h_{L,enc}^{t-1}$ to turn t.

We see that explicitly adding skip-connections substantially improves performance in EQM, from 0 or 6% to 55%, and has a positive effect on BLEU. The models show similar behavior on CS_small. In this case, when an api_call is executed, the result is treated as a response and sent as input to the next turn. Although Model 4 performed the best

Table 2: Results with variants of the seq2seq model on the bAbI dataset.

Model	Type	Description	BLEU	P	Acc	EQM
Model 1	Basic Seq2Seq	dependencies between turns absent	88.3	0.60	0.87	0.00
Model 2	Skip connection	append $h^{t-1}_{L',dec}$	90.2	1.00	1.00	0.06
Model 3	Seq2Seq	Model 1 with an attention layer	93.4	1.00	1.00	0.26
Model 4	Skip connection	Model 2 + $h^{t-1}_{L,enc}$	95.8	1.00	1.00	0.55

Table 3: Results with the Nearest Neighbor approach on customer support data (CS_small).

Model	Description	BLEU	P	R	Acc	EQM
Model 4	Skip connection	9.91	0.34	0.79	0.81	0.30
Model 6	Nearest Neighbor using Word2Vec	11.06	0.31	0.24	0.86	0.10
Model 7	Nearest Neighbor using Sent2Vec	14.39	0.29	0.26	0.85	0.09
Model 8	Nearest Neighbor using discounted Sent2Vec	16.43	0.56	0.60	0.91	0.21
Model 9	Nearest neighbor using output of encoder	15.14	0.38	0.35	0.86	0.13
Model 10	Nearest neighbor using output of decoder	16.34	0.36	0.31	0.86	0.16
Model 11	Best Of both (Models 4+10)	17.67	0.33	0.73	0.80	0.30

on CS_small and CS_large, our analysis showed that the generated responses were most often incoherent and not fluent, a phenomenon that did not arise in the synthetic dataset. We now proceed to explain the nearest neighbor based approach, which we show is able to produce reasonable responses that are more fluent.

3.2 Nearest Neighbor-based approach

In our nearest neighbor approach, an agent's response is chosen from human generated transcripts or the training data - ensuring fluency. However, this does not necessarily ensure that the responses are coherent in the context of the dialogue. The nearest neighbor approach starts with a representation of the entire dialogue history $bs_{t,i}$ for turn t and dialogue i. Together with $a_{t,i}$, the action the agent took while in this state i.e., the natural language response or api_call query issued by the agent, this results in a tuple $< bs_{t,i}, a_{t,i} >$. The entire training data is converted into a set of tuples S, that contains pairwise relationships between dialog state representations and agent actions.

In the online or test phase, given an embedding of the dialogue so far, $testVec$, we find the nearest neighbor $bs_{testVec}$ in S. We return the nearest neighbor's corresponding response, $a_{testVec}$, as the predicted agent's response. We use ball trees (Kibriya and Frank, 2007) to perform efficient nearest neighbor search. Since we want to provide more flexibility to the human agent in choosing the most appropriate response, we extended this approach to find $k = 100$ responses and then used a diversity-based ranking approach (Zhu et al., 2007) to return 5 diverse responses. To construct the adjacency matrix for diversity ranking, we use word overlap between responses after stop word removal.

Numerous techniques have been proposed for representating text including word2vec and sent2vec (Mikolov et al., 2013b,a; Pagliardini et al., 2017; Pennington et al., 2014). In the following sections, we compare these approaches against our proposed representations using skip connections.

3.2.1 Dialogue Embeddings from Word/Sentence Embeddings

In our first baseline, **Model 6**, for a dialogue, i, the user's response at turn t, $user_t$, is concatenated with his/her responses in previous turns ($user_{i,1:t-1}$) and the agent's responses upto turn $t-1$ ($agent_{i,1:t-1}$), to obtain, $p_{i,t} = (user_{i,1:t}, agent_{i,1:t-1})$. We obtain a belief state vector representation as the average of the word2vec (Mikolov et al., 2013b) representations of words in $p_{i,t}$. We then apply the nearest neighbor approach described in Section 3.2. Results obtained with this approach on CS_small are in Table 3.

We emphasize a subtle but important oracle advantage that we give this baseline algorithm. When we obtain the embeddings of a test dialogue, we use the true utterances of the expert agent so far,

Table 4: Results with the Nearest Neighbor approach on customer support data (CS_large).

Model	Description	BLEU	Online-BLEU
Model 4	Skip connection	8.9	46.2
Model 5	Lucene	8.4	39.2
Model 10	Nearest neighbor using output of decoder	13.5	90.8

which would not be available in practice. However, we will show that our proposed representation, described in Section 3.3, performs better, even without access to this information.

Pagliardini et al. (2017) recently described a method that leads to better sentence-level representations. We use their approach as another baseline. bs_t is represented by the average of the sentence embeddings of all agent's responses upto turn $t-1$ and user's responses upto turn t. We also explore geometric discounting to give higher importance to recent responses. We use a similar process to obtain representations for the user's responses during the test phase. As done with word-embeddings, we provide true agent responses upto turn $t-1$ for predicting the agent's response at turn t. Results obtained on CS_small by averaging (**Model 7**) and discounted averaging (**Model 8**) are given in Table 3. Model 8 performs better than Model 7 across all measures. A comparison between Model 6, 7 and 8 with Model 4 in Table 3, would not be a fair one as Model 4 does not use previous *true* agent responses to predict the agent's next response.

3.3 Hybrid model: Nearest Neighbor with Seq2Seq Embeddings

We suggest using the outputs of the hidden units in the decoder of our skip connection seq2seq model, as suitable representations for the belief states. The seq2seq model for handling multi-turn dialogue is trained as before (Section 3.1.1). Once the parameters have been learned, we proceed to generate representations for all turns in the training data. The output of the last hidden unit of the encoder or the decoder before turn t is used to represent the belief state vector at turn t. As before, we obtain a set S consisting of pairs of belief state vectors and next actions taken by the agent.

We test the models as done in Section 3.1.1, except now we select responses using the nearest neighbor approach (Figure 2). Results obtained are in Table 3 (Models 9 and 10). **Model 9** uses the output of the last hidden unit of the encoder. **Model 10** uses previous turn's decoder's last hidden unit. Both the models show a significant improvement in BLEU when compared to generating the agent's response (Model 4). Although Model 10 was not exposed to the past true agent responses, it still achieved comparable performance to that of Model 8. Appending both the encoder and the decoder outputs did not have significant impact.

The results also show that the seq2seq model achieved a better EQM when compared to the nearest neighbor approach. The final hybrid model, we propose (**Model 11**) combines both strategies. We run both the Models 4 and 10 in parallel. When Model 4 predicts an API response, we use the output generated by Model 4 as the agent's response, otherwise we use the output of Model 10 as the predicted agent's response. This model achieved the best results among all models we study, both in terms of fluency (BLEU) as well as correctness of external actions (EQM). The hybrid model achieves a 78% relative improvement (from 9.91 to 17.67) in fluency scores, and 200% improvement in EQM over previous approaches (from 0.10 to 0.30).

Table 4 shows results obtained on CS_large (column 3) using models that performed the best on the other datasets. Another obvious baseline is to use traditional retrieval approaches. (query, agent response) pairs are created for each agent response, with a query constructed by concatenating all the agent's responses upto turn $t-1$ and user's responses upto turn t, for an agent response at time t. For a given dialogue history query, the corresponding agent response is retrieved using Lucene[3]. Since CS_large did not contain labeled api_calls, we report results using Model 10. As seen, Model 10 provides a substantial boost in performance.

3.4 Manual Online Evaluation

One caveat to the above evaluations is that they are based on customer responses to the actual human agent interactions, and are not fully indicative of how customers would react to the real automated system in practice. Another disadvantage of using

[3] https://lucene.apache.org/

Table 5: Sample responses show interesting human behaviors learned by the approach.

Example	User's Utterance	Agent's Response
1	YES YES YES , get it done .	done :)
	perfect	sir , thanks for waiting
2	it was clearly your fault	i understand sir
	when should i return it and how	you can keep it or you can donate or else you can dispose it off .

automated evaluation with just one reference, is that the score (BLEU) penalizes valid responses that may be lexically different from the available agent response. To overcome this issue, we conducted online experiments with human agents.

We used 5 human users and 2 agents. On average each user interacted with an agent on 10 different issues that needed resolution. To compare against our baseline, each user interacted with the Model 4, 5 and 10 using the same issues. This resulted in ≈ 50 dialogues from each of the models. After every response from the user, the human agent was allowed to select one of the top five responses the system selected. We refer to the selected response as A. The human agent was asked to make minimal modifications to the selected response, resulting in a response A'. If the responses suggested were completely irrelevant, the human agent was allowed to type in the most suitable response.

We then computed the BLEU between the system generated responses (As) and human generated responses (A's), referred to as Online-BLEU in Table 4. Since the human agent only made minimal changes where appropriate, we believe the BLEU score would now be more correlated to human judgments. Since CS_large did not contain any api_calls, we only report BLEU scores. The results obtained with models 4, 5 and 10 on CS_large are shown in Table 4 (column 4). Model 10 performs better than Models 4 and 5. We do not measure inter-annotator agreement as each human user can take a different dialog trajectory.

We noticed that the approach mimics certain interesting human behavior. For example, in Table 5, the chatbot detects that the user is frustrated and responds with smileys and even makes exceptions on the return policy.

4 Conclusion and Future Work

We demonstrated limitations of previous end-end dialog approaches and proposed variants to make them suitable for real world settings. In ongoing work, we explore reinforcement learning techniques to reach the goal state quicker thereby reducing the number of interactions.

References

S. Bennacef, L. Devillers, S. Rosset, and L. Lamel. 1996. Dialog in the railtel telephone-based system. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*. volume 1, pages 550–553 vol.1.

Antoine Bordes and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *CoRR* abs/1605.07683.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR* .

Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander H. Miller, Arthur Szlam, and Jason Weston. 2015. Evaluating prerequisite qualities for learning end-to-end dialog systems. *CoRR* abs/1511.06931.

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

Ashraf M. Kibriya and Eibe Frank. 2007. An empirical comparison of exact nearest neighbour algorithms. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*. Springer-Verlag, Berlin, Heidelberg, PKDD 2007, pages 140–151.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *CoRR* abs/1706.03872.

Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *CoRR* abs/1603.08023.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. Curran Associates Inc., USA, NIPS'13, pages 3111–3119.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *CoRR* abs/1703.02507.

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*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '02, pages 311–318.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1532–1543.

Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2015. A survey of available corpora for building data-driven dialogue systems. *CoRR* abs/1512.05742.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short text conversation. *ACL* .

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015a. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, New York, NY, USA, CIKM '15, pages 553–562.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015b. A neural network approach to context-sensitive generation of conversational responses. *CoRR* abs/1506.06714.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. *CoRR* abs/1506.05869.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. *EACL* .

Jason D. Williams and Geoffrey Zweig. 2016. End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. *CoRR* abs/1606.01269.

Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, SIGIR '16, pages 55–64.

Tiancheng Zhao and Maxine Eskénazi. 2016. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. *CoRR* abs/1606.02560.

Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. 2016. Multi-view response selection for human-computer conversation. In *EMNLP*. pages 372–381.

Xiaojin Zhu, Andrew Goldberg, Jurgen Van Gael, and David Andrzejewski. 2007. Improving diversity in ranking using absorbing random walks. *HLT-NAACL* pages 97–104.

Victor Zue, Stephanie Seneff, James Glass, Joseph Polifroni, Christine Pao, Timothy J. Hazen, and Lee Hetherington. 2000. Jupiter: A telephone-based conversational interface for weather information. *IEEE Trans. on Speech and Audio Processing* 8:85–96.