Exploring Personalized Neural Conversational Models

Satwik Kottur[1], **Xiaoyu Wang**[2], **Vitor R. Carvalho**[2]
[1]Carnegie Mellon University, Pittsburgh, PA
[2]Snap Inc., Venice, CA
skottur@andrew.cmu.edu,{xiaoyu.wang,vitor}@snap.com

Abstract

Modeling dialog systems is currently one of the most active problems in Natural Language Processing. Recent advances in Deep Learning have sparked an interest in the use of neural networks in modeling language, particularly for personalized conversational agents that can retain contextual information during dialog exchanges. This work carefully explores and compares several of the recently proposed neural conversation models, and carries out a detailed evaluation on the multiple factors that can significantly affect predictive performance, such as pretraining, embedding training, data cleaning, diversity-based reranking, evaluation setting, etc. Based on the tradeoffs of different models, we propose a new neural generative dialog model conditioned on speakers as well as context history that outperforms previous models on both retrieval and generative metrics. Our findings indicate that pretraining speaker embeddings on larger datasets, as well as bootstrapping word and speaker embeddings, can significantly improve performance (up to 3 points in perplexity), and that promoting diversity in using Mutual Information based techniques has a very strong effect in ranking metrics.

1 Introduction

Modeling dialog systems is one of the most active and problems in natural language processing. Successful dialog systems highlight our ability to replicate complete language understanding and thus clear the 'Turing Test', a true test for machine intelligence. The recent advancement in deep learning has sparked an interest in use of neural networks in modeling language. In particular, dialog or conversational models too have received a lot of attention due to its wide range of applications in human-machine interaction such as personal assistants, technical support for products and services, entertainment, to name a few [Lowe *et al.*, 2015; Serban *et al.*, 2016a; Wen *et al.*, 2015; Li *et al.*, 2016b; Sordoni *et al.*, 2015].

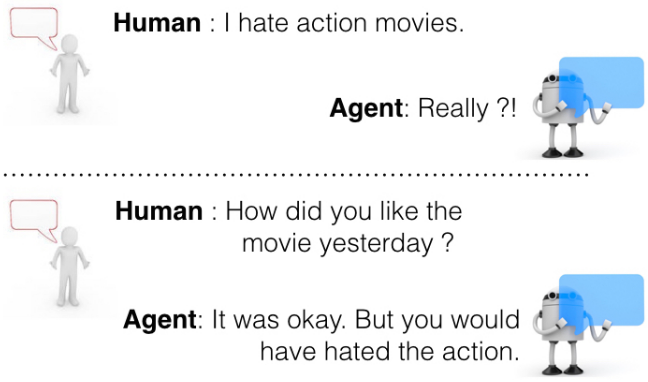
Human : I hate action movies.
Agent: Really ?!
Human : How did you like the movie yesterday ?
Agent: It was okay. But you would have hated the action.

Figure 1: Illustration of personalized and context-aware dialogue system. The agent generates responses that are personalized to the specific user and in the right context.

In contrast to classical rule-based models, deep learning models leverage huge amounts of language data and can in principle be trained end-to-end. Recently proposed models [Sordoni *et al.*, 2015; Li *et al.*, 2016b; Serban *et al.*, 2016b] have shown success in training neural dialog systems that return semantic and syntactic responses for a given input. However, several challenges such as enforcing consistency, context-awareness and personalization still remain largely unsolved. Also, it is relatively unclear the extent to which various components of these models contribute to the overall prediction quality.

To help address these issues, we present below the two main contributions of this paper.

First, we carefully compare several of the recently proposed neural conversation models, and carry out a detailed evaluation on the multiple factors that can significantly affect predictive performance, such as pretraining, embedding training, data cleaning, diversity reranking, evaluation setting (retrieval or generative evaluation), etc.

Second, we propose a new generative dialog model conditioned on both speakers and context history that outperforms previous models on generative metrics (such as Perplexity) as well as on retrieval metrics (such as Recall@K).

Our findings indicate that pretraining speaker embeddings on larger datasets, as well as bootstrapping word and speaker embeddings, can significantly boost predictive performance (up to 3 points in perplexity), and that promoting diversity via Mutual Information based techniques has a strong effect in ranking metrics.

The paper is organized as follow: Sec. 2 discusses some of related work in conversational agents. We introduce our model along with other baselines in Sec. 3, followed by description of the datasets used in Sec. 4. Our experimental details including training procedure, model variants and evaluation, are elaborated in Sec. 5. Finally, we present our results in Sec. 6 and conclude with discussions in Sec. 7.

2 Related Work

Conversation exchanges have traditionally been modeled using heuristics, templates, hand-crafted rules or statistically learning parts of (a usually complex) dialog system from relatively small amounts of data. However, the recent availability of large amounts of conversation data has opened the gates for the creation of several data-driven dialog models. For instance, using millions of Twitter conversation exchanges, Ritter et al. [Ritter *et al.*, 2011] modeled dialog responses as generated from dialog questions using a phrase-based statistical machine translation system. One of the main advantages of data-driven models is that they can be created in an end-to-end manner, that is, purely derived from its input data and with no explicit bias or assumptions on dialog structure.

In addition to the availability of large dialog collections, recent advances in Deep Learning have led to significant improvements on several NLP tasks. Works on distributed representation of language [Mikolov and Dean, 2013], neural language models [Bengio *et al.*, 2003] and sequence-to-sequence learning [Sutskever *et al.*, 2014] have significantly changed the state-of-the-art landscape in NLP. One of the first attempts of neural dialog models was proposed by Vinyals and Le [Vinyals and Le, 2015], where a dialog response is generated from a dialog question (or previous sentence) in a sequence-to-sequence framework. Further improvements on this generative modeling idea were later explored in [Li *et al.*, 2016b], [Sordoni *et al.*, 2015], [Serban *et al.*, 2015a] and [Galley *et al.*, 2015].

Instead of generating sentences in dialogs, another plausible approach to conversation modeling is to retrieve the best possible answer from a large collection of previous dialogs. For instance, Ameixa et al [Ameixa *et al.*, 2014] used movie subtitles to retrieve good answers to out-of-domain questions. More recently, Al-Rfou et al. [Al-Rfou *et al.*, 2016] utilized an enormous amount of dialog data derived from Reddit and created a personalized and contextualized neural ranking model able to retrieve the best answers from a large collection. Kannan et al. [Kannan *et al.*, 2016] proposed a smart reply system for email where relevant pre-selected answers are ranked according to the incoming message contents.

Our proposed model draws inspiration from the ideas of [Serban *et al.*, 2015a] and [Al-Rfou *et al.*, 2016], but extends them to contextualize in a generative model both *personas* as well as previous dialog contexts. More importantly, we provide a comprehensive evaluation, revisiting various assumptions on the training procedure of these models (and its components), and how each one affect their final performance in both generation and retrieval settings.

3 Models

In this section, we first introduce notation and briefly overview GRU [Cho *et al.*, 2014], the underlying recurrent neural network (RNN) for all the models. We then discuss some baselines for our work that either do not consider one of context and personalization or both. Finally, we describe our proposed conversational model that conditions on both the history and speaker information.

3.1 Notation

Models in this work are trained using a dataset of conversations with multiple participant speakers (see Sec. 4 for details). A conversation C_i is an ordered set of pairs of turn D_i^j and speaker of the turn S_i^j, i.e., $C_i = \{(D_i^1, S_i^1), (D_i^2, S_i^2), \cdots, (D_i^n, S_i^n)\}$. Each turn itself is a collection of words uttered by the corresponding speaker, $D_i^j = \{w_{i1}^j, w_{i2}^j, \cdots, w_{in}^j\}$. For sake of brevity, we ignore i indexing the conversations. All our language models predict the distribution of the next word conditioned on previous information at each time step. The difference in the extent of information used from previous turns gives rise to different models that we aim to compare in this work.

3.2 Gated Recurrent Unit (GRU)

GRU [Cho *et al.*, 2014] is used to model a sequence of inputs, e.g., words in a sentence in our case. For a sentence $W = \{w_1, w_2, \cdots, w_n\}$, a GRU associates hidden state h_t for each time step. At every time step, the hidden state from previous time step h_{t-1} and current member from the sequence w_t form the input. These inputs are used to evolve the hidden state to give h_t, by means of update and reset gates denoted by z_t and r_t respectively. Finally, the evolved hidden state at current time step h_t is also used to predict the distribution of the next word w'_{t+1} over a vocabulary through a softmax transformation. With $\sigma(.)$ being the sigmoid function, the update equations are given as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, w_t]) \quad (1)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, w_t]) \quad (2)$$

$$\tilde{h_t} = tanh\,(W \cdot [r_t * h_{t-1}, w_t]) \quad (3)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h_t} \quad (4)$$

$$w'_{t+1} = softmax\,(W_o \cdot h_t) \quad (5)$$

where $*$ represents element-wise multiplication and w's are embeddings for the corresponding words. Matrices W_z, W_r, W, W_o are parameters learnt by optimizing a loss function for given train data, depending on the task at hand. In a simple setting, one could maximize the probability of observed next word as predicted by GRU.

3.3 Baselines

For quantitative comparison, we train the following neural models as baselines. Additionally, we also consider a statistical language model, an n-gram (where $n = 5$) model with Kneser-Ney smoothing [Kneser and Ney, 1995].

Encoder-Decoder: Neural conversational model [Vinyals and Le, 2015] couples two RNNs, an encoder and decoder, and learns to predict a novel response for a given input sentence. This approach does not consider speaker information and is dyadic in nature, i.e., the next response is dependent only on the current dialog and independent of other information. We call this `enc-dec` baseline, and the language model is shown below:

$$P(w_{t+1}^j | w_1^j, \cdots, w_t^j, D^{-j}, S^{-j}, S^j) \\ = P(w_{t+1}^j | w_1^j, \cdots, w_t^j, D^{j-1}) \quad (6)$$

Here, D^{-j} denotes the set of all previous turns upto j, i.e., $D^{-j} = \{D^1, D^2, \cdots, D^{j-1}\}$. Similarly for speakers S^{-j}.

Persona-only: Next we consider the persona-based conversational model [Li *et al.*, 2016b] that extends the basic encoder-decoder model by including speaker as an additional input for both encoder and decoder RNNs. In particular, we consider the speaker-addressee model of [Li *et al.*, 2016b] and call it `persona`. The distribution of next word at a given time step is as follows:

$$P(w_{t+1}^j | w_1^j, \cdots, w_t^j, D^{-j}, S^{-j}, S^j) \\ = P(w_{t+1}^j | w_1^j, \cdots, w_t^j, D^{j-1}, S^{j-1}, S^j) \quad (7)$$

Thus, the language model is dependent on the speaker of the current and previous turns, achieving personalization while generating novel responses to sentences.

Context-only: The Hierarchical Recurrent Encoder-Decoder [Serban *et al.*, 2016b] captures contextual cues through a high level, context RNN that updates its hidden for every turn in a conversation and is learnt on top of encoder-decoder RNNs. However, it does not encapsulate any personalization information and thus forms the `context` baseline for our experiments. For details, we refer the reader to [Serban *et al.*, 2015a]. The language model in this case is as follows:

$$P(w_{t+1}^j | w_1^j, \cdots, w_t^j, D^{-j}, S^{-j}, S^j) \\ = P(w_{t+1}^j | w_1^j, \cdots, w_t^j, D^{-j}) \quad (8)$$

Notice the dependence of the next word on all of the previous turns, thereby retaining the context information through history. However, in practice, history is truncated and only past few turns are considered.

3.4 CoPerHED Model

Our model **Co**ntext-aware, **Per**sona-based **H**ierarchical **E**ncoder-**D**ecoder (CoPerHED) is a hybrid of persona-based [Li *et al.*, 2016b] and context-aware [Serban *et al.*, 2016b] neural conversational models. By combining the two, we condition the language model both on the context of conversation as well as the speaker information for current and previous turns, thus achieving the desired personalization while generating novel responses. The proposed model architecture is depicted in Fig. 2. Akin to [Li *et al.*, 2016b], speaker information for the current and previous turns is fed into the RNN as an additional input. Context-awareness is achieved similar to the hierarchical model in [Serban *et al.*, 2016b], where an additional RNN summarizes information at the level of turns. Thus, CoPerHED has access to entire history both in terms of context and speaker persona while predicting the next word. The language model is therefore dependent on sentences and speakers in previous turns along with current speaker. We now detail three constituents:

Encoder: An encoder is simply a RNN used to convert a sentence into an encoded vector. The hidden state of encoder RNN for j^{th} turn h_t^j is evolved by feeding one word at each time step along with current speaker annotation. Representations for all the words and speakers are jointly learnt as embeddings in the model. The final hidden state h_n^j after processing all the words in the sentence is used as a representation for the entire sentence, and forms the input to the context RNN as shown in Fig. 2. If s^j denotes the speaker embedding for the current turn, then:

$$h_t^j = GRU(h_{t-1}^j, \left[w_t^j, s^j\right]) \quad (9)$$

where $GRU(.)$ denotes the GRU function (see Sec. 3.2). Notice that the speaker persona s^j is fixed for all time steps for the given turn.

Context RNN: Encoded representations for sentences at each turn in the conversation is processed by the context RNN. This helps retain relevant information from the previous turns and serves as context for predicting responses. Similar to [Serban *et al.*, 2016b], CoPerHED constitutes a hierarchy of RNNs with context RNN working at a turn-level in a conversation while language RNN (GRU in this case) works at a word-level in every turn. Let g_j denote the hidden state of context RNN at j^{th} turn, then:

$$g_j = GRU(g_{j-1}, h_n^j) \quad (10)$$

Decoder: Similar to encoder, decoder RNN processes words one at a time to predict the distribution of the next word. The current word, speaker for the current turn and the hidden state from context RNN are its inputs. Using $\hat{h}_t$ to represent the hidden state of the decoder, we have:

$$\hat{h}_t^{j+1} = GRU(\hat{h}_{t-1}^{j+1}, \left[w_t, s^{j+1}, g_j\right]) \quad (11)$$

The probability of next word is given from $\hat{h}_t^{j+1}$ similar to Eq. 5. Training is carried out by maximizing the loglikehood of w_t^{j+1} predicted by the decoder.

4 Datasets

To evaluate our method, we need a dataset with everyday, free form natural language conversations between multiple people. Such a dataset would help capture and condition the language model on both desired aspects–context of conversation and persona of speaker. To this end, we choose three datasets comprising of subtitles from famous movies and TV shows as they are good sources for *teaching and learning spoken language features* [Forchini, 2009].

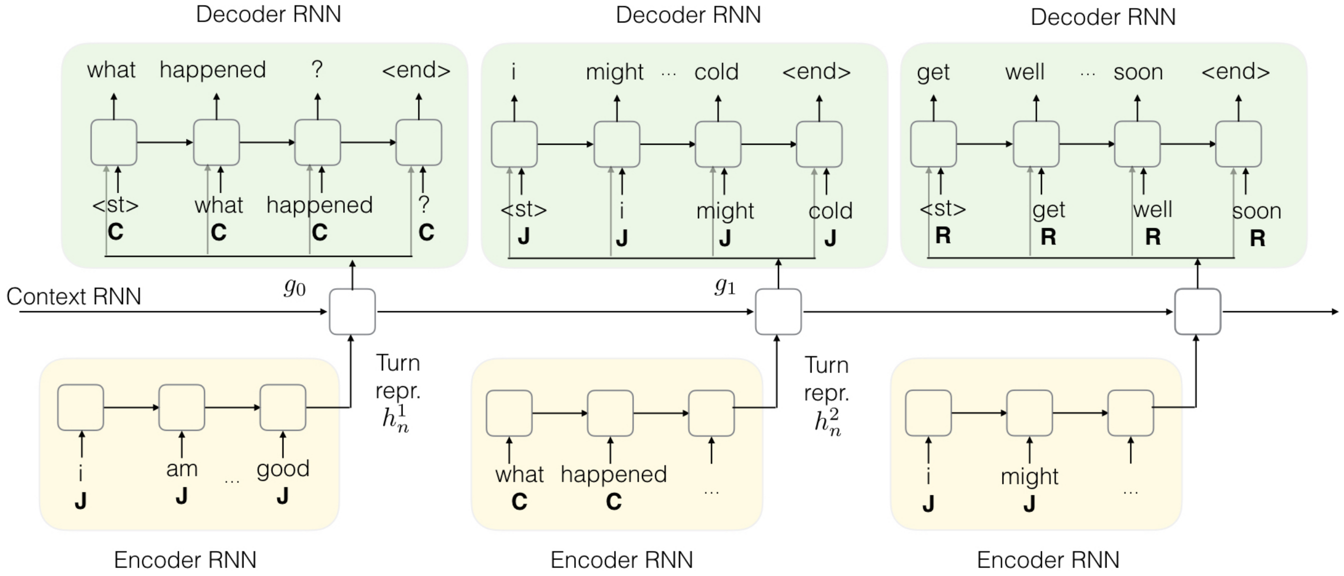

Figure 2: Architecture of CoPerHED explained with an example. Speakers are labeled with single letter for brevity. *Joey: i am not feeling good.; Chandler: what happened?; Joey: i might have cold.; Ross: get well soon*. The encoder RNN admits an additional input for speaker. The final hidden state of encoder, used as representation for the turn, is processed by context RNN. The output of context RNN is fed into decoder RNN along with current word and next speaker, to generate novel responses.

Movie-DiC dataset: The *Movie-DiC* dataset [Banchs, 2012] was collected through The Internet Movie Script Data Collection (IMSDb)[1], which contains publicly available movie subtitles. The dataset has around 132K conversations from 753 movies that comprises of roughly 764K turns. It has explicit speaker annotations and retains the order of turns in a given conversation making it a good choice to train our model.

TV-Series dataset: We use freely available transcripts for two American television comedy shows, Friends[2] and The Big Bang Theory[3], to construct our TV-Series dataset. We parse the corresponding HTML pages to extract all the turns preserving the conversation structure and speaker annotations. There are around 4.4K conversations with a total of 93K speaker turns in this dataset. Due to the small size, we do not train directly on this dataset but instead finetune a model pre-trained on a larger dataset to avoid overfitting.

SubTle dataset The third dataset we consider is the SubTle [Ameixa *et al.*, 2014], which is an enormous, non-dialog corpus extracted from movie subtitles. It has around 5.5M pairs of turns without speaker annotations and hence is not favorable to directly train our model. However, we use it to pre-train giving us a strong language prior as discussed in Sec. 5.3.

	Movie-DiC	TV-Series	SubTle
Total convs.	38.3K	4.40K	5.50M
Total turns	437K	93.2K	11.0M
Tokens	6.00M	1.29M	96.5M
Speakers	1.79K	60	-
Turns/speaker	3.35K	21.6K	-

Table 1: Dataset statistics after preprocessing (Sec. 5.1).

5 Experiments

We now explain the setup to train and evaluate our conversational model that is context-aware and conditions on the persona of the speaker.

5.1 Data Preprocessing

The datasets (Sec. 4) are preprocessed as follows. We follow [Serban *et al.*, 2015b] and tag named entities using the NER tagger from the standard NLTK library [Bird *et al.*, 2009] and replace them with placeholders (e.g. <*PERSON*>, <*PLACE*>, etc.). Next, we split the dataset into three non-overlapping partitions – train (80%), validation (10%) and test (10%) We then construct our vocabulary by considering only the words that occur at least 10 times in the dataset. All of the remaining words are replaced with an unknown token (<*UNK*>). Similarly, speakers who have at least 50 turns in the training set are considered while the remaining are mapped to an unknown speaker (<*UNS*>). Table 1 shows the statistics after dataset preprocessing. TO be able to generalize, we use the same vocabulary of size $10.4K$ obtained from train data of Movie-DiC, for all the datasets.

[1] http://www.imsdb.com/

[2] http://transcripts.foreverdreaming.org/viewforum.php?f=159

[3] https://bigbangtrans.wordpress.com/

Model	smp	w2v	spk	bth	pre
Kneser-Ney			56.58		
`enc-dec`	42.80	41.35	-	-	35.22
`persona`	42.74	41.43	42.16	40.89	35.01
`context`	42.30	41.15	-	-	33.95
`CoPerHED`	**41.35**	**40.56**	**41.07**	**39.82**	**33.66**

Table 2: Perplexity for various models in the generative task on Movie-DiC dataset. Lower the better. CoPerHED outperforms other baselines in all the settings (std: ± 0.1).

5.2 Training

We use the deep learning framework Torch [Collobert *et al.*, 2011], to build and train our models by minimizing the log-likelihood of tokens predicted by the language model. To perform fair comparisons across all the model architectures, we use 2-layered Gated Recurrent Unit (GRU) for both the encoder and decoder with a dropout [Srivastava *et al.*, 2014] of 0.2. The parameters of the network are learnt through standard back-propagation algorithm with adam optimizer [Kingma and Ba, 2014]. The learning rate is set to 0.001 and is decayed exponentially to 0.0001 by the end of 10 epochs, after which it is held constant. Gradient values are clipped to within $[-5.0, 5.0]$ to avoid explosion. Training is terminated once the perplexity on the held-out validation set saturates for the given model. To find the best hyperparameters, we once again use perplexity on validation set across various settings. For all our models, best performance was achieved when:

- Word embeddings size is 300
- Speaker embedding size is 50
- Number of hidden units for encoder/decoder GRU is 300
- Number of hidden units for context GRU is 50

5.3 Initialization

Neural models have high learning capacity with a lot of parameters and tend to overfit easily if the size of training dataset is not sufficiently large. Therefore, it is usually advantageous to learn priors either by bootstrapping few parameters or pre-training using a larger general corpus. We experiment with three such settings as explained below:

Bootstrap word embeddings: As discussed in Sec. 3, the model jointly learns semantic representations for one-hot encoded words along with the parameters, using the training dataset. Instead of training them from scratch, one could initialize with representations learnt offline. To this end, we use word2vec [Mikolov and Dean, 2013] trained on a huge corpus of 1 billion words as these capture rich notions of semantic relatedness useful for language models. Specifically, we use the 300-dimension, publicly available [4] embeddings.

Bootstrap speaker embeddings: Similar to word embeddings, models dependent on persona learn vector representations for one-hot encoded speakers. Thus, apart from learning them from scratch we also experiment initializing these speaker embeddings with hand-crafted features. In particular, we construct bag-of-words (BOW) feature for all the speakers from training data and reduce its dimension as needed using PCA. It is interesting to observe that even a simple BOW feature improves over random initialization.

Pre-train on SubTle dataset: The performance of a neural model usually gets better with increasing the amount of training data. However, it is not always trivial to find large datasets of required nature either due to expensive annotations or inherent unavailability. In such cases, previous works have shown success while pre-training using an other huge dataset and then fine-tuning on a desired target dataset for the actual task at hand. Following this intuition, we additionally pre-train our models on a much larger SubTle dataset before fine-tuning on the target dataset, which is either the Movie-DiC or TV-Series dataset. As SubTle dataset is neither speaker-annotated nor has entire conversations, we simply assign all speakers to $<$*UNS*$>$ and treat randomly chosen lines as a conversation for pre-training.

Model Variants: For each language model, we learn variants based on how the weights have been initialized:

- **smp**: trained with random initialization,
- **w2v**: bootstrap word embeddings using word2vec,
- **spk**: bootstrap speaker embeddings using BOW,
- **bth**: bootstrap both the embeddings,
- **pre**: use SubTle pre-trained initialization.

5.4 Evaluation

We consider two different evaluation setups as given below:

Generation: Automatic evaluation of probabilistic language models that generate novel responses remains an open problem of research [Liu *et al.*, 2016]. Though it is desired that generated language be sound both syntactically and semantically, its relevance cannot be easily evaluated. Metrics like Perplexity, BLEU, and deltaBLEU, have been adopted from machine translation to understand the performance of such generative models. Following [Serban *et al.*, 2016b], we use perplexity in our study. Perplexity, a measure of likelihood of an unseen test set according to a model, has been used with success in various tasks such as machine translation, image captioning and speech recognition.

Retrieval: Additionally, we consider retrieval as accurate metrics that reflect performance can be used. To evaluate for a turn in a conversation, we consider an answer pool of size N containing the ground truth and $N-1$ other, randomly sampled turns from the test set. The generative language model is used to score all the sentences from the answer pool and re-rank according to a scoring function. We then use **recall@k** metric that measures the percent of sentences whose ground truth was ranked in the top-K of the answer pool. We discuss the choice of scoring function in the next subsection. It must be noted that all the neural models are trained with a

[4] `https://code.google.com/archive/p/word2vec/`

Model	smp		w2v		spk		bth	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
Random	10.00	50.00	10.00	50.00	10.00	50.00	10.00	50.00
`enc-dec`	36.66	82.14	38.23	83.77	-	-	-	-
`persona`	**37.93**	83.51	**38.95**	84.79	**40.02**	85.48	**41.28**	87.21
`context`	34.92	85.34	38.16	85.11	-	-	-	-
`CoPerHED`	37.57	**85.64**	36.25	**86.02**	37.92	**87.02**	39.40	**88.06**

Table 3: Retrieval results for Movie-DiC dataset with $N = 10$. Metrics used are Recall@1 and Recall@5, higher the better. Our CoPerHED improves over other models for R@5. std=±0.35%

Model	Perplexity
Kneser-Ney	71.77
`enc-dec`	34.20
`persona`	34.63
`context`	32.08
`CoPerHED`	**31.48** (-8%)

Table 4: Perplexity on TV show dataset. Our model CoPerHED shows an improvement of 8%. We only finetune models on this dataset due to small size.

generative loss and therefore are not optimized for retrieval. However, such a retrieval setup would help us understand the discriminative nature of the model and thereby evaluate their ability to capture the underlying language distribution.

5.5 Promoting diversity using MI

Generative language models tend to produce safe and generic responses, e.g., *I don't know*, *Yes* and so on. High frequency of such phrases in the train dataset could be a potential reason for such a phenomenon [Serban *et al.*, 2016b]. Recent works [Li *et al.*, 2016a; Wen *et al.*, 2015] mitigate this by either using a diversity promoting objective or re-ranking multiple responses. Following [Li *et al.*, 2016a], we adopt Mutual Information as our scoring function in the retrieval setup. We observed that changing this scoring function from Likelihood to Mutual Information resulted in drastic performance improvement for all the models alike.

6 Results

6.1 Generative

Tab. 2 summarizes the generative results on the Movie-DiC dataset. Firstly, all the neural models outperform Kneser-Ney smoothing by more than 14 perplexity points. This is intuitive as statistical methods suffer from the curse of dimensionality with increase in size of data. CoPerHED outperforms all the competing baselines in any variant, thus confirming our hypothesis that context-awareness and personalization leads to better conversational modeling. We also see that bootstrapping with either word (w2v) or speaker (spk) embeddings improves performance for all models alike. In particular, word2vec initialization is more effective as it is pre-trained on Google billion word corpus [Chelba *et al.*, 2013] thereby providing a rich prior. Interestingly, initializing both embeddings (bth) outperforms either one (w2v, spk), suggesting a complementary benefit obtained from each source. However, the best performance in each model is obtained by pretraining on SubTle dataset that is atleast one order of magnitude larger. Pretraining resulted in a drop of atleast 7 perplexity points. This behavior is as expected because neural networks allow successful transfer learning, i.e., pretrain on a huge corpus before fine-tuning on desired, target corpus.

Results for generative setup on TV series is given in Tab. 4. We again find that CoPerHED outperforms all the baselines by around 3 perplexity points resulting in an improvement of 8% over `enc-dec`.

6.2 Retrieval

We use $N = 10$ for our experiments on retrieval. Note that we re-use generative models for retrieval task and do not train them discriminatively. Clearly, CoPerHED has the best Recall@5 metric amongst all the competing methods. However, `persona` seems to outperform other models in terms of Recall@1. We hypothesize that this could be due to stronger persona-based cues while retrieving from an answer pool of $N = 10$. To summarize, language prediction in CoPerHED takes advantage of both personalization and context-awareness, leading to performance improvement over baselines, which we expect to increase further with larger datasets.

7 Discussion and Conclusion

In this work, we attempted to better understand generative neural conversation models. We described some of the key components of such models, and carefully investigated issues that can severely affect model performance, including pretraining strategies, speaker and word embeddings initializations, and data cleaning conventions. Using different datasets we observed consistently that pretraining speaker embeddings on larger datasets, as well as bootstrapping word and speaker embeddings appropriately, can have a very positive and noticeable impact on performance.

We also proposed a new neural context-aware and personalized generative dialog model (CoPerHED) that, by jointly accounting for persona and previous dialog context, was able to outperform previous baselines on multiple datasets. We also evaluated all models on a retrieval setting (under the same assumptions), where we observed that promoting diversity via Mutual Information based techniques has a very strong effect in ranking metrics.

As future work, we intend to further investigate the tradeoffs between a retrieval versus a generation setting for dialog modeling. Different settings may be used for different situations. Another future research direction lies on the cold start problem for a brand new speaker. That is, how to better model a new speaker in a well known dialog context.

References

[Al-Rfou *et al.*, 2016] Rami Al-Rfou, Marc Pickett, Javier Snaider, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. Conversational contextual cues: The case of personalization and history for response ranking. *arXiv preprint arXiv:1606.00372*, 2016.

[Ameixa *et al.*, 2014] David Ameixa, Luisa Coheur, Pedro Fialho, and Paulo Quaresma. *Luke, I am Your Father: Dealing with Out-of-Domain Requests by Using Movies Subtitles*, pages 13–21. Cham, 2014.

[Banchs, 2012] Rafael E. Banchs. Movie-DiC: a movie dialogue corpus for research and development. In *ACL*, pages 203–207, Jeju Island, Korea, July 2012.

[Bengio *et al.*, 2003] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *JMLR*, 3(Feb):1137–1155, 2003.

[Bird *et al.*, 2009] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[Chelba *et al.*, 2013] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. Technical report, Google, 2013.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, Doha, Qatar, October 2014.

[Collobert *et al.*, 2011] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011.

[Forchini, 2009] P. Forchini. Spontaneity reloaded: American face-to-face and movie conversation compared. In *Corpus Linguistics 2009. Abstracts of the Corpus Linguistics Conference*, 2009.

[Galley *et al.*, 2015] Michel Galley, Chris Brockett, Alessandro Sordoni, Yangfeng Ji, Michael Auli, Chris Quirk, Margaret Mitchell, Jianfeng Gao, and Bill Dolan. deltableu: A discriminative metric for generation tasks with intrinsically diverse targets. In *ACL*, pages 445–450, Beijing, China, July 2015.

[Kannan *et al.*, 2016] Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufman, Balint Miklos, Greg Corrado, Andrew Tomkins, Laszlo Lukacs, Marina Ganea, Peter Young, and Vivek Ramavajjala. Smart reply: Automated response suggestion for email. In *KDD (2016).*, 2016.

[Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[Kneser and Ney, 1995] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. *ICASSP*, 1:181–184, 1995.

[Li *et al.*, 2016a] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *NAACL*, pages 110–119, 2016.

[Li *et al.*, 2016b] Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In *ACL*, pages 994–1003, 2016.

[Liu *et al.*, 2016] Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*, 2016.

[Lowe *et al.*, 2015] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*, 2015.

[Mikolov and Dean, 2013] T Mikolov and J Dean. Distributed representations of words and phrases and their compositionality. *Advances in NIPS*, 2013.

[Ritter *et al.*, 2011] Alan Ritter, Colin Cherry, and William B Dolan. Data-driven response generation in social media. In *EMNLP*, pages 583–593, 2011.

[Serban *et al.*, 2015a] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Hierarchical neural network generative models for movie dialogues. *arXiv preprint arXiv:1507.04808*, 2015.

[Serban *et al.*, 2015b] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. Hierarchical neural network generative models for movie dialogues. *CoRR*, abs/1507.04808, 2015.

[Serban *et al.*, 2016a] Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. *arXiv preprint arXiv:1603.06807*, 2016.

[Serban *et al.*, 2016b] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. *arXiv preprint arXiv:1605.06069*, 2016.

[Sordoni *et al.*, 2015] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. In *NAACL*, pages 196–205, Denver, Colorado, May–June 2015.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in NIPS*, pages 3104–3112, 2014.

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

[Wen *et al.*, 2015] Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *arXiv preprint arXiv:1508.01755*, 2015.