A Persona-Based Neural Conversation Model

Jiwei Li[1*] **Michel Galley**[2] **Chris Brockett**[2]
Georgios P. Spithourakis[3*] **Jianfeng Gao**[2] **Bill Dolan**[2]

[1]Stanford University, Dept. of Computer Science
jiweil@stanford.edu
[2]Microsoft Research, Redmond
{mgalley,chrisbkt,jfgao,billdol}@microsoft.com
[3]University College London, Dept. of Computer Science
g.spithourakis@cs.ucl.ac.uk

Abstract

We present persona-based models for handling the issue of speaker consistency in neural response generation. A speaker model encodes personas in distributed embeddings that capture individual characteristics such as background information and speaking style. A dyadic speaker-addressee model captures properties of interactions between two interlocutors. Our models yield qualitative performance improvements in both perplexity and BLEU scores over baseline sequence-to-sequence models, with similar gains in speaker consistency as measured by human judges.

1 Introduction

As conversational agents gain traction as user interfaces, there has been growing research interest in training naturalistic conversation systems from large volumes of human-to-human interactions (Ritter et al., 2011; Sordoni et al., 2015; Vinyals and Le, 2015; Li et al., 2016). One major issue for these data-driven systems is their propensity to select the response with greatest likelihood—in effect a consensus response of the humans represented in the training data. Outputs are frequently vague or non-committal (Li et al., 2016), and when not, they can be wildly inconsistent, as illustrated in Table 1.

In this paper, we address the challenge of consistency and how to endow data-driven systems with the coherent "persona" needed to model human-like behavior, whether as personal assistants, personalized avatar-like agents, or game characters.[1] For present purposes, we will define PERSONA as the character that an artificial agent, as actor, plays or performs during conversational interactions. A persona can be viewed as a composite of elements of identity (background facts or user profile), language behavior, and interaction style. A persona is also adaptive, since an agent may need to present different facets to different human interlocutors depending on the interaction.

Fortunately, neural models of conversation generation (Sordoni et al., 2015; Shang et al., 2015; Vinyals and Le, 2015; Li et al., 2016) provide a straightforward mechanism for incorporating personas as embeddings. We therefore explore two per-

message	Where do you live now?
response	I live in Los Angeles.
message	In which city do you live now?
response	I live in Madrid.
message	In which country do you live now?
response	England, you?
message	Where were you born?
response	I was born in Canada.
message	Where are you from?
response	England, you?
message	Where did you grow up?
response	I grew up in Texas.
message	How old are you?
response	16 and you?
message	What's your age?
response	18.
message	What is your major?
response	I'm majoring in psychology
message	What did you study in college?
response	English lit.

Table 1: Inconsistent responses generated by a 4-layer SEQ2SEQ model trained on 25 million Twitter conversation snippets.

* The entirety of this work was conducted at Microsoft.

[1](Vinyals and Le, 2015) suggest that the lack of a coherent personality makes it impossible for current systems to pass the Turing test.

sona models, a single-speaker SPEAKER MODEL and a dyadic SPEAKER-ADDRESSEE MODEL, within a sequence-to-sequence (SEQ2SEQ) framework (Sutskever et al., 2014). The Speaker Model integrates a speaker-level vector representation into the target part of the SEQ2SEQ model. Analogously, the Speaker-Addressee model encodes the interaction patterns of two interlocutors by constructing an interaction representation from their individual embeddings and incorporating it into the SEQ2SEQ model. These persona vectors are trained on human-human conversation data and used at test time to generate personalized responses. Our experiments on an open-domain corpus of Twitter conversations and dialog datasets comprising TV series scripts show that leveraging persona vectors can improve relative performance up to 20% in BLEU score and 12% in perplexity, with a commensurate gain in consistency as judged by human annotators.

2 Related Work

This work follows the line of investigation initiated by Ritter et al. (2011) who treat generation of conversational dialog as a statistical machine translation (SMT) problem. Ritter et al. (2011) represents a break with previous and contemporaneous dialog work that relies extensively on hand-coded rules, typically either building statistical models on top of heuristic rules or templates (Levin et al., 2000; Young et al., 2010; Walker et al., 2003; Pieraccini et al., 2009; Wang et al., 2011) or learning generation rules from a minimal set of authored rules or labels (Oh and Rudnicky, 2000; Ratnaparkhi, 2002; Banchs and Li, 2012; Ameixa et al., 2014; Nio et al., 2014; Chen et al., 2013). More recently (Wen et al., 2015) have used a Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) to learn from unaligned data in order to reduce the heuristic space of sentence planning and surface realization.

The SMT model proposed by Ritter et al., on the other hand, is end-to-end, purely data-driven, and contains no explicit model of dialog structure; the model learns to converse from human-to-human conversational corpora. Progress in SMT stemming from the use of neural language models (Sutskever et al., 2014; Gao et al., 2014; Bahdanau et al., 2015; Luong et al., 2015) has inspired efforts to extend these neural techniques to SMT-based conversational response generation. Sordoni et al. (2015) augments Ritter et al. (2011) by rescoring outputs using a SEQ2SEQ model conditioned on conversation history. Other researchers have recently used SEQ2SEQ to directly generate responses in an end-to-end fashion without relying on SMT phrase tables (Serban et al., 2015; Shang et al., 2015; Vinyals and Le, 2015). Serban et al. (2015) propose a hierarchical neural model aimed at capturing dependencies over an extended conversation history. Recent work by Li et al. (2016) measures mutual information between message and response in order to reduce the proportion of generic responses typical of SEQ2SEQ systems. Yao et al. (2015) employ an intention network to maintain the relevance of responses.

Modeling of users and speakers has been extensively studied within the standard dialog modeling framework (e.g., (Wahlster and Kobsa, 1989; Kobsa, 1990; Schatztnann et al., 2005; Lin and Walker, 2011)). Since generating meaningful responses in an open-domain scenario is intrinsically difficult in conventional dialog systems, existing models often focus on generalizing character style on the basis of qualitative statistical analysis (Walker et al., 2012; Walker et al., 2011). The present work, by contrast, is in the vein of the SEQ2SEQ models of Vinyals and Le (2015) and Li et al. (2016), enriching these models by training persona vectors directly from conversational data and relevant side-information, and incorporating these directly into the decoder.

3 Sequence-to-Sequence Models

Given a sequence of inputs $X = \{x_1, x_2, ..., x_{n_X}\}$, an LSTM associates each time step with an input gate, a memory gate and an output gate, respectively denoted as i_t, f_t and o_t. We distinguish e and h where e_t denotes the vector for an individual text unit (for example, a word or sentence) at time step t while h_t denotes the vector computed by the LSTM model at time t by combining e_t and h_{t-1}. c_t is the cell state vector at time t, and σ denotes the sigmoid function. Then, the vector representation h_t for each time step t is given by:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ l_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \cdot \begin{bmatrix} h_{t-1} \\ e_t^s \end{bmatrix} \quad (1)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot l_t \quad (2)$$

$$h_t^s = o_t \cdot \tanh(c_t) \quad (3)$$

where W_i, W_f, W_o, $W_l \in \mathbb{R}^{K \times 2K}$. In SEQ2SEQ generation tasks, each input X is paired with a sequence of outputs to predict: $Y = \{y_1, y_2, ..., y_{n_Y}\}$. The LSTM defines a distribution over outputs and sequentially predicts tokens using a softmax function:

$$p(Y|X) = \prod_{t=1}^{n_y} p(y_t|x_1, x_2, ..., x_t, y_1, y_2, ..., y_{t-1})$$
$$= \prod_{t=1}^{n_y} \frac{\exp(f(h_{t-1}, e_{y_t}))}{\sum_{y'} \exp(f(h_{t-1}, e_{y'}))}$$

where $f(h_{t-1}, e_{y_t})$ denotes the activation function between h_{t-1} and e_{y_t}. Each sentence terminates with a special end-of-sentence symbol *EOS*. In keeping with common practices, inputs and outputs use different LSTMs with separate parameters to capture different compositional patterns.

During decoding, the algorithm terminates when an *EOS* token is predicted. At each time step, either a greedy approach or beam search can be adopted for word prediction.

4 Personalized Response Generation

Our work introduces two persona-based models: the Speaker Model, which models the personality of the respondent, and the Speaker-Addressee Model which models the way the respondent adapts their speech to a given addressee — a linguistic phenomenon known as lexical entrainment (Deutsch and Pechmann, 1982).

4.1 Notation

For the response generation task, let M denote the input word sequence (message) $M = \{m_1, m_2, ..., m_I\}$. R denotes the word sequence in response to M, where $R = \{r_1, r_2, ..., r_J, EOS\}$ and J is the length of the response (terminated by an *EOS* token). r_t denotes a word token that is associated with a K dimensional distinct word embedding e_t. V is the vocabulary size.

4.2 Speaker Model

Our first model is the Speaker Model, which models the respondent alone. This model represents each individual speaker as a vector or embedding, which encodes speaker-specific information (e.g., dialect, register, age, gender, personal information) that influences the content and style of her responses. Note that these attributes are not explicitly annotated, which would be tremendously expensive for our datasets. Instead, our model manages to cluster users along some of these traits (e.g., age, country of residence) based on the responses alone.

Figure 1 gives a brief illustration of the Speaker Model. Each speaker $i \in [1, N]$ is associated with a user-level representation $v_i \in \mathbb{R}^{K \times 1}$. As in standard SEQ2SEQ models, we first encode message S into a vector representation h_S using the source LSTM. Then for each step in the target side, hidden units are obtained by combining the representation produced by the target LSTM at the previous time step, the word representations at the current time step, and the speaker embedding v_i:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ l_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \cdot \begin{bmatrix} h_{t-1} \\ e_t^s \\ v_i \end{bmatrix} \quad (4)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot l_t \quad (5)$$

$$h_t^s = o_t \cdot \tanh(c_t) \quad (6)$$

where $W \in \mathbb{R}^{4K \times 3K}$. In this way, speaker information is encoded and injected into the hidden layer at each time step and thus helps predict personalized responses throughout the generation process. The Speaker embedding $\{v_i\}$ is shared across all conversations that involve speaker i. $\{v_i\}$ are learned by back propagating word prediction errors to each neural component during training.

Another useful property of this model is that it helps *infer* answers to questions even if the evidence is not readily present in the training set. This is important as the training data does not contain explicit information about every attribute of each user (e.g., gender, age, country of residence). The model learns speaker representations based on conversational content produced by different speakers, and speakers producing similar responses tend to have similar embeddings, occupying nearby positions in the vector space. This way, the training data of speakers nearby in vector space help increase the generalization capability of the speaker model. For example, consider two speakers i and j who sound distinctly British, and who are therefore close in speaker embedding space. Now, suppose that, in the training data, speaker i was asked *Where do you live?* and responded *in the UK*. Even if speaker j was never asked the same question, this answer can help influence a good response from speaker j, and this without explicitly labeled geo-location information.

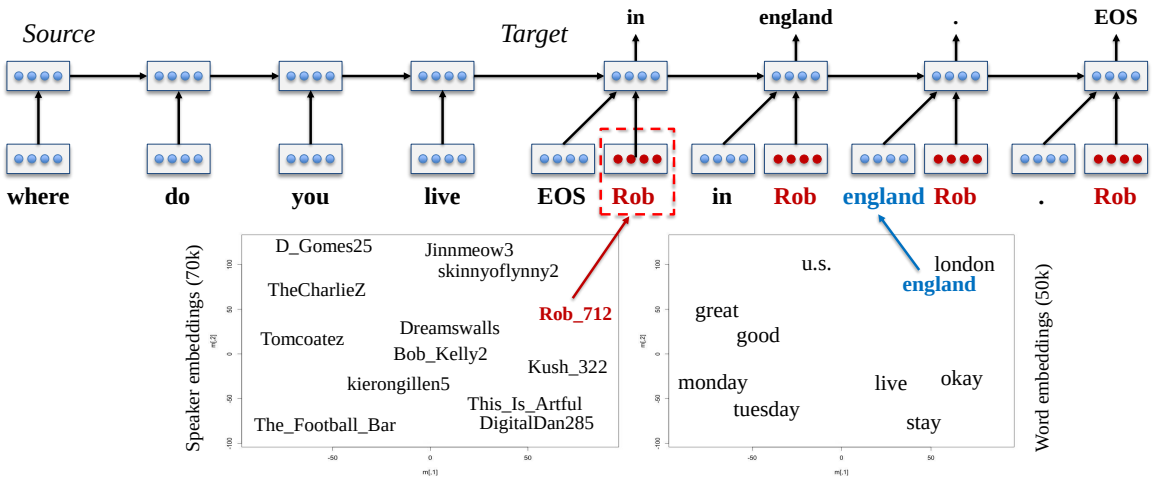

Figure 1: Illustrative example of the Speaker Model introduced in this work. Speaker IDs close in embedding space tend to respond in the same manner. These speaker embeddings are learned jointly with word embeddings and all other parameters of the neural model via backpropagation. In this example, say Rob is a speaker clustered with people who often mention England in the training data, then the generation of the token 'england' at time $t = 2$ would be much more likely than that of 'u.s.'. A non-persona model would prefer generating *in the u.s.* if 'u.s.' is more represented in the training data across all speakers.

4.3 Speaker-Addressee Model

A natural extension of the Speaker Model is a model that is sensitive to speaker-addressee interaction patterns within the conversation. Indeed, speaking style, register, and content does not vary only with the identity of the speaker, but also with that of the addressee. For example, in scripts for the TV series *Friends* used in some of our experiments, the character Ross often talks differently to his sister Monica than to Rachel, with whom he is engaged in an on-again off-again relationship throughout the series.

The proposed Speaker-Addressee Model operates as follows: We wish to predict how speaker i would respond to a message produced by speaker j. Similarly to the Speaker model, we associate each speaker with a K dimensional speaker-level representation, namely v_i for user i and v_j for user j. We obtain an interactive representation $V_{i,j} \in \mathbb{R}^{K\times 1}$ by linearly combining user vectors v_i and v_j in an attempt to model the interactive style of user i towards user j,

$$V_{i,j} = \tanh(W_1 \cdot v_i + W_2 \cdot v_2) \tag{7}$$

where $W_1, W_2 \in \mathbb{R}^{K\times K}$. $V_{i,j}$ is then linearly incorporated into LSTM models at each step in the target:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ l_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \cdot \begin{bmatrix} h_{t-1} \\ e^s_t \\ V_{i,j} \end{bmatrix} \tag{8}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot l_t \tag{9}$$

$$h^s_t = o_t \cdot \tanh(c_t) \tag{10}$$

$V_{i,j}$ depends on both speaker and addressee and the same speaker will thus respond differently to a message from different interlocutors. One potential issue with Speaker-Addressee modelling is the difficulty involved in collecting a large-scale training dataset in which each speaker is involved in conversation with a wide variety of people. Like the Speaker Model, however, the Speaker-Addressee Model derives generalization capabilities from speaker embeddings. Even if the two speakers at test time (i and j) were never involved in the same conversation in the training data, two speakers i' and j' who are respectively close in embeddings may have been, and this can help modelling how i should respond to j.

4.4 Decoding and Reranking

For decoding, the N-best lists are generated using the decoder with beam size $B = 200$. We set a maximum length of 20 for the generated candidates. Decoding operates as follows: At each time step, we first examine all $B \times B$ possible next-word candidates, and add all hypothesis ending with an *EOS* token to the N-best list. We then preserve the top-B unfinished hypotheses and move to the next word position.

To deal with the issue that SEQ2SEQ models tend to generate generic and commonplace responses such as *I don't know*, we follow Li et al. (2016) by reranking the generated N-best list using

a scoring function that linearly combines a length penalty and the log likelihood of the source given the target:

$$\log p(R|M, v) + \lambda \log p(M|R) + \gamma |R| \quad (11)$$

where $p(R|M, v)$ denotes the probability of the generated response given the message M and the respondent's speaker ID. $|R|$ denotes the length of the target and γ denotes the associated penalty weight. We optimize γ and λ on N-best lists of response candidates generated from the development set using MERT (Och, 2003) by optimizing BLEU. To compute $p(M|R)$, we train an inverse SEQ2SEQ model by swapping messages and responses. We trained standard SEQ2SEQ models for $p(M|R)$ with no speaker information considered.

5 Datasets

5.1 Twitter Persona Dataset

Data Collection Training data for the Speaker Model was extracted from the Twitter FireHose for the six-month period beginning January 1, 2012. We limited the sequences to those where the responders had engaged in at least 60 (and at most 300) 3-turn conversational interactions during the period, in other words, users who reasonably frequently engaged in conversation. This yielded a set of 74,003 users who took part in a minimum of 60 and a maximum of 164 conversational turns (average: 92.24, median: 90). The dataset extracted using responses by these "conversationalists" contained 24,725,711 3-turn sliding-window (context-message-response) conversational sequences.

In addition, we sampled 12000 3-turn conversations from the same user set from the Twitter FireHose for the three-month period beginning July 1, 2012, and set these aside as development, validation, and test sets (4000 conversational sequences each). Note that development, validation, and test sets for this data are single-reference, which is by design. Multiple reference responses would typically require acquiring responses from different people, which would confound different personas.

Training Protocols We trained four-layer SEQ2SEQ models on the Twitter corpus following the approach of (Sutskever et al., 2014). Details are as follows:

- 4 layer LSTM models with 1,000 hidden cells for each layer.
- Batch size is set to 128.
- Learning rate is set to 1.0.
- Parameters are initialized by sampling from the uniform distribution $[-0.1, 0.1]$.
- Gradients are clipped to avoid gradient explosion with a threshold of 5.
- Vocabulary size is limited to 50,000.
- Dropout rate is set to 0.2.

Source and target LSTMs use different sets of parameters. We ran 14 epochs, and training took roughly a month to finish on a Tesla K40 GPU machine.

As only speaker IDs of responses were specified when compiling the Twitter dataset, experiments on this dataset were limited to the Speaker Model.

5.2 Twitter Sordoni Dataset

The Twitter Persona Dataset was collected for this paper for experiments with speaker ID information. To obtain a point of comparison with prior state-of-the-art work (Sordoni et al., 2015; Li et al., 2016), we measure our baseline (non-persona) LSTM model against prior work on the dataset of (Sordoni et al., 2015), which we call the Twitter Sordoni Dataset. We only use its test-set portion, which contains responses for 2114 context and messages. It is important to note that the Sordoni dataset offers up to 10 references per message, while the Twitter Persona dataset has only 1 reference per message. Thus BLEU scores cannot be compared across the two Twitter datasets (BLEU scores on 10 references are generally much higher than with 1 reference). Details of this dataset are in (Sordoni et al., 2015).

5.3 Television Series Transcripts

Data Collection For the dyadic Speaker-Addressee Model we used scripts from the American television comedies *Friends*[2] and *The Big Bang Theory*,[3] available from Internet Movie Script Database (IMSDb).[4] We collected 13 main characters from the two series in a corpus of 69,565 turns. We split the corpus into training/development/testing sets, with development and testing sets each of about 2,000 turns.

Training Since the relatively small size of the dataset does not allow for training an open domain dialog model, we adopted a domain adaption strategy where we first trained a standard SEQ2SEQ

[2] https://en.wikipedia.org/wiki/Friends
[3] https://en.wikipedia.org/wiki/The_Big_Bang_Theory
[4] http://www.imsdb.com

System	BLEU
MT baseline (Ritter et al., 2011)	3.60%
Standard LSTM MMI (Li et al., 2016)	5.26%
Standard LSTM MMI (our system)	5.82%
Human	*6.08%*

Table 2: BLEU on the Twitter Sordoni dataset (10 references). We contrast our baseline against an SMT baseline (Ritter et al., 2011), and the best result (Li et al., 2016) on the established dataset of (Sordoni et al., 2015). The last result is for a human oracle, but it is not directly comparable as the oracle BLEU is computed in a leave-one-out fashion, having one less reference available. We nevertheless provide this result to give a sense that these BLEU scores of 5-6% are not unreasonable.

models using a much larger OpenSubtitles (OSDb) dataset (Tiedemann, 2009), and then adapting the pre-trained model to the TV series dataset.

The OSDb dataset is a large, noisy, open-domain dataset containing roughly 60M-70M scripted lines spoken by movie characters. This dataset does not specify which character speaks each subtitle line, which prevents us from inferring speaker turns. Following Vinyals et al. (2015), we make the simplifying assumption that each line of subtitle constitutes a full speaker turn.[5] We trained standard SEQ2SEQ models on OSDb dataset, following the protocols already described in Section 5.1. We run 10 iterations over the training set.

We initialize word embeddings and LSTM parameters in the Speaker Model and the Speaker-Addressee model using parameters learned from OpenSubtitles datasets. User embeddings are randomly initialized from $[-0.1, 0.1]$. We then ran 5 additional epochs until the perplexity on the development set stabilized.

6 Experiments

6.1 Evaluation

Following (Sordoni et al., 2015; Li et al., 2016) we used BLEU (Papineni et al., 2002) for parameter tuning and evaluation. BLEU has been shown to correlate well with human judgment on the response generation task, as demonstrated in (Galley et al., 2015). Besides BLEU scores, we also report perplexity as an indicator of model capability.

6.2 Baseline

Since our main experiments are with a new dataset (the Twitter Persona Dataset), we first show that our LSTM baseline is competitive with the state-of-the-art (Li et al., 2016) on an established dataset, the Twitter Sordoni Dataset (Sordoni et al., 2015). Our baseline is simply our implementation of the LSTM-MMI of (Li et al., 2016), so results should be relatively close to their reported results. Table 2 summarizes our results against prior work. We see that our system actually does better than (Li et al., 2016), and we attribute the improvement to a larger training corpus, the use of dropout during training, and possibly to the "conversationalist" nature of our corpus.

[5]This introduces a degree of noise as consecutive lines are not necessarily from the same scene or two different speakers.

Model	Standard LSTM	Speaker Model
Perplexity	47.2	42.2 (-10.6%)

Table 3: Perplexity for standard SEQ2SEQ and the Speaker model on the Twitter Persona development set.

Model	Objective	BLEU
Standard LSTM	MLE	0.92%
Speaker Model	MLE	1.12% (+21.7%)
Standard LSTM	MMI	1.41%
Speaker Model	MMI	1.66% (+11.7%)

Table 4: BLEU on the Twitter Persona dataset (1 reference), for the standard SEQ2SEQ model and the Speaker model using as objective either maximum likelihood (MLE) or maximum mutual information (MMI).

6.3 Results

We first report performance on the Twitter Persona dataset. Perplexity is reported in Table 3. We observe about a 10% decrease in perplexity for the Speaker model over the standard SEQ2SEQ model. In terms of BLEU scores (Table 4), a significant performance boost is observed for the Speaker model over the standard SEQ2SEQ model, yielding an increase of 21% in the maximum likelihood (MLE) setting and 11.7% for mutual information setting (MMI). In line with findings in (Li et al., 2016), we observe a consistent performance boost introduced by the MMI objective function over a standard SEQ2SEQ model based on the MLE objective function. It is worth noting that our persona models are more beneficial to the MLE models than to the MMI models. This result is intuitive as the persona models help make Standard LSTM MLE outputs more informative and less bland, and thus make the use of MMI less critical.

For the TV Series dataset, perplexity and BLEU scores are respectively reported in Table 5 and Table 6. As can be seen, the Speaker and Speaker-Addressee models respectively achieve perplexity values of 25.4 and 25.0 on the TV-series dataset,

Model	Standard LSTM	Speaker Model	Speaker-Addressee Model
Perplexity	27.3	25.4 (−7.0%)	25.0 (−8.4%)

Table 5: Perplexity for standard SEQ2SEQ and persona models on the TV series dataset.

Model	Standard LSTM	Speaker Model	Speaker-Addressee Model
MLE	1.60%	1.82% (+13.7%)	1.83% (+14.3%)
MMI	1.70%	1.90% (+10.6%)	1.88% (+10.9%)

Table 6: BLEU on the TV series dataset (1 reference), for the standard SEQ2SEQ and persona models.

7.0% and 8.4% percent lower than the correspondent standard SEQ2SEQ models. In terms of BLEU score, we observe a similar performance boost as on the Twitter dataset, in which the Speaker model and the Speaker-Addressee model outperform the standard SEQ2SEQ model by 13.7% and 10.6%. By comparing the Speaker-Addressee model against the Speaker model on the TV Series dataset, we do not observe a significant difference. We suspect that this is primarily due to the relatively small size of the dataset where the interactive patterns might not be fully captured. Smaller values of perplexity are observed for the Television Series dataset than the Twitter dataset, the perplexity of which is over 40, presumably due to the noisier nature of Twitter dialogues.

6.4 Qualitative Analysis

Diverse Responses by Different Speakers Table 7 represents responses generated by persona models in response to three different input questions. We randomly selected 10 speakers (without cherry-picking) from the original Twitter dataset. We collected their user level representations from a speaker look-up table and integrated them into the decoding models. The model tends to generate specific responses for different people in response to the factual questions.[6]

Table 8 shows responses generated from the *Speaker-Addressee Model* using the TV-series dataset. Interestingly, we regularly observe that this model is sensitive to the identity of the addressee, generating words specifically targeted at that addressee (e.g., her name). For example, the model produces *Of course, I love you, Emily* in response to an input from *Emily*. Also, the model generates *Of course I love you. (kisses him)*, where the pronoun "him" accurately identifies the gender of the addressee.

[6]There appears to be a population bias in the training set that favors British users.

message	Where is your hometown?
baseline	I was born in Canada.
user1	I'm from England.
user2	I'm from Manchester.
user3	I'm from Liverpool.
user4	England. You?
user5	My hometown.
user6	I'm from Texas.
user7	I'm from LA.
user8	England.
user9	I was born here.
user10	I was born in the us.
message	What company do you work for?
baseline	I work for a company.
user1	I don't have a job.
user2	I'm a manager.
user3	I'm working. #
user4	I work for a company.
user5	I'm working. #
user6	Customer service.
user7	I work in retail.
user8	I work in a cafe.
user9	I work at a gas station.
user10	I don't know #
message	What did you have for dinner?
baseline	I had fish and chips.
user1	I had spag bol.
user2	Chicken and chips.
user3	Chicken and rice.
user4	Fish and chips.
user5	I had spag bol.
user6	I had Mexican food.
user7	Salad...
user8	I had chicken and chips.
user9	I had spag bol.
user10	Pizza.

Table 7: Responses generated by the baseline (LSTM-MMI) and the Speaker Model for ten randomly selected users, without cherry picking. # indicates poor-quality responses produced by the system.

Human Evaluation We conducted a human evaluation of outputs from the Speaker Model, using a crowdsourcing service. Since we cannot expect crowdsourced human judges to know or attempt to learn the ground truth of Twitter users who are not well-known public figures, we designed our experiment to evaluate the consistency of outputs associated with the speaker IDs. To this end, we collected 24 pairs of questions for which we would expect

message	Do you love me?
baseline	Of course, I love you.
	addressee: Emily, speaker: Sheldon
response	Of course, I love you, Emily.
	addressee: Emily, speaker: Penny
response	Of course, I love you, Emily.
	addressee: Leonard, speaker: Penny
response	Of course, I love you, Leonard.
	addressee: Monica, speaker: Chandler
response	Of course I love you. I want to marry you.
	addressee: Rachel, speaker: Chandler
response	Of course I love you.
	addressee: Ross, speaker: Chandler
response	Of course I love you.
	addressee: Ross, speaker: Rachel
response	Of course I love you. (kisses him)
	addressee: Emily, speaker: Rachel
response	Of course I love you.

Table 8: Responses to *Do you love me?* from the *Speaker-Addressee* model on the TV-series dataset using different addressees and speakers.

responses to be consistent if the persona model is coherent. For example, responses to the questions *What country do you live in?* and *What city do you live in?* would be considered consistent if the answers were *England* and *London* respectively, but not if they were *UK* and *Chicago*. Similarly, the responses to *Are you vegan or vegetarian?* and *Do you eat beef?* are consistent if the answers generated are *vegan* and *absolutely not*, but not if they are *vegan* and *I love beef*. We collected 20 pairs of outputs for randomly-selected personas provided by the Speaker Model for each question pair (480 response pairs total). We also obtained the corresponding outputs from the baseline MMI-enhanced SEQ2SEQ system.

Since our purpose is to measure the gain in consistency over the baseline system, we presented the pairs of answers system-pairwise, i.e., 4 responses, 2 from each system, displayed on the screen, and asked judges to decide which of the two systems was more consistent. The position in which the system pairs were presented on the screen was randomized. The two systems were judged on 5-point zero-sum scale, assigning a score of 2 (-2) if one system was judged more (less) consistent than the other, and 1 (-1) if one was rated "somewhat" more (less) consistent. Ties were assigned a score of zero. Five judges rated each pair and their scores were averaged and remapped into 5 equal-width bins. After discarding ties, we found the persona model was judged either "more consistent" or "somewhat more consistent" in 56.7% of cases. If we ignore the "somewhat more consistent" judgments, the persona model wins in 6.1% of cases, compared with only 1.6% for the baseline model. It should be emphasized that the baseline model is a strong baseline, since it represents the consensus of all 70K Twitter users in the dataset[7].

User1	
message	Where is your hometown?
response	I'm from England.
message	Where are you from?
response	I'm from England.
message	In which city do you live now?
response	I live in London.
message	In which country do you live now?
response	I live in England.
message	What is your major?
response	Business. You?
message	What did you study in college?
response	I did business studies.
message	How old are you?
response	I'm 18.
message	What is your age?
response	I'm 18.
User2	
message	Where is your hometown?
response	I live in Indonesia.
message	Where are you from?
response	England, you?
message	In which city do you live now?
response	I live in Jakarta.
message	In which country do you live now?
response	I live in Indonesia.
message	What is your major?
response	Business, you?
message	What did you study in college?
response	Psychology, you?
message	How old are you?
response	I'm 18.
message	What is your age?
response	I'm 16.

Table 9: Examples of speaker consistency and inconsistency generated by the Speaker Model

Table 9 illustrates how consistency is an emergent property of two arbitrarily selected users. The model is capable of discovering the relations between different categories of location such as London and the UK, Jakarta and Indonesia. However, the model also makes inconsistent response decisions, generating different answers in the second example in response to questions asking about age or major. Our proposed persona models integrate user embeddings into the LSTM, and thus can be viewed as encapsulating a trade-off between a persona-specific generation model and a general conversational model.

[7] *I'm not pregnant* is an excellent consensus answer to the question *Are you pregnant?*, while *I'm pregnant* is consistent as a response only in the case of someone who also answers the question *Are you a guy or a girl?* with something in the vein of *I'm a girl*.

7 Conclusions

We have presented two persona-based response generation models for open-domain conversation generation. There are many other dimensions of speaker behavior, such as mood and emotion, that are beyond the scope of the current paper and must be left to future work.

Although the gains presented by our new models are not spectacular, the systems outperform our baseline SEQ2SEQ systems in terms of BLEU, perplexity, and human judgments of speaker consistency. We have demonstrated that by encoding personas in distributed representations, we are able to capture personal characteristics such as speaking style and background information. In the Speaker-Addressee model, moreover, the evidence suggests that there is benefit in capturing dyadic interactions.

Our ultimate goal is to be able to take the profile of an arbitrary individual whose identity is not known in advance, and generate conversations that accurately emulate that individual's persona in terms of linguistic response behavior and other salient characteristics. Such a capability will dramatically change the ways in which we interact with dialog agents of all kinds, opening up rich new possibilities for user interfaces. Given a sufficiently large training corpus in which a sufficiently rich variety of speakers is represented, this objective does not seem too far-fetched.

Acknowledgments

We with to thank Stephanie Lukin, Pushmeet Kohli, Chris Quirk, Alan Ritter, and Dan Jurafsky for helpful discussions.

References

David Ameixa, Luisa Coheur, Pedro Fialho, and Paulo Quaresma. 2014. Luke, I am your father: dealing with out-of-domain requests by using movies subtitles. In *Intelligent Virtual Agents*, pages 13–21. Springer.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of the International Conference on Learning Representations (ICLR)*.

Rafael E Banchs and Haizhou Li. 2012. IRIS: a chat-oriented dialogue system based on the vector space model. In *Proc. of the ACL 2012 System Demonstrations*, pages 37–42.

Yun-Nung Chen, Wei Yu Wang, and Alexander Rudnicky. 2013. An empirical investigation of sparse log-linear models for improved dialogue act classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8317–8321. IEEE.

Werner Deutsch and Thomas Pechmann. 1982. Social interaction and the development of definite descriptions. *Cognition*, 11:159–184.

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

Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. 2014. Learning continuous phrase representations for translation modeling. In *Proc. of ACL*, pages 699–709, Baltimore, Maryland.

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

Alfred Kobsa. 1990. User modeling in dialog systems: Potentials and hazards. *AI & society*, 4(3):214–231.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1):11–23.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proc. of NAACL-HLT*.

Grace I Lin and Marilyn A Walker. 2011. All the world's a stage: Learning character models from film. In *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proc. of ACL*, pages 11–19, Beijing, China, July.

Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, Mirna Adriani, and Satoshi Nakamura. 2014. Developing non-goal dialog system based on examples of drama television. In *Natural Interaction with Robots, Knowbots and Smartphones*, pages 355–361. Springer.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July. Association for Computational Linguistics.

Alice H Oh and Alexander I Rudnicky. 2000. Stochastic language generation for spoken dialogue systems. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3*, pages 27–32.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318.

Roberto Pieraccini, David Suendermann, Krishna Dayanidhi, and Jackson Liscombe. 2009. Are we there yet? research in commercial spoken dialog systems. In *Text, Speech and Dialogue*, pages 3–13. Springer.

Adwait Ratnaparkhi. 2002. Trainable approaches to surface natural language generation and their application to conversational dialog systems. *Computer Speech & Language*, 16(3):435–455.

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 583–593.

Jost Schatztnann, Matthew N Stuttle, Karl Weilhammer, and Steve Young. 2005. Effects of the user model on simulation-based learning of dialogue strategies. In *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, pages 220–225.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2015. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proc. of AAAI*.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *ACL-IJCNLP*, pages 1577–1586.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Meg Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proc. of NAACL-HLT*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems (NIPS)*, pages 3104–3112.

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

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *Proc. of ICML Deep Learning Workshop*.

Wolfgang Wahlster and Alfred Kobsa. 1989. *User models in dialog systems*. Springer.

Marilyn A Walker, Rashmi Prasad, and Amanda Stent. 2003. A trainable generator for recommendations in multimodal dialog. In *INTERSPEECH*.

Marilyn A Walker, Ricky Grant, Jennifer Sawyer, Grace I Lin, Noah Wardrip-Fruin, and Michael Buell. 2011. Perceived or not perceived: Film character models for expressive nlg. In *Interactive Storytelling*, pages 109–121. Springer.

Marilyn A Walker, Grace I Lin, and Jennifer Sawyer. 2012. An annotated corpus of film dialogue for learning and characterizing character style. In *LREC*, pages 1373–1378.

William Yang Wang, Ron Artstein, Anton Leuski, and David Traum. 2011. Improving spoken dialogue understanding using phonetic mixture models. In *FLAIRS Conference*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proc. of EMNLP*, pages 1711–1721, Lisbon, Portugal, September. Association for Computational Linguistics.

Kaisheng Yao, Geoffrey Zweig, and Baolin Peng. 2015. Attention with intention for a neural network conversation model. *CoRR*, abs/1510.08565.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.