# Improving the Dialogue Generation Consistency via Self-supervised Learning

Yizhe Zhang Xiang Gao Sungjin Lee Chris Brockett Michel Galley Jianfeng Gao Bill Dolan Microsoft Research, Redmond, WA, USA

{yizzhang, xiag, sule, chrisbkt, mgalley, jfgao, billdol}@microsoft.com

#### **Abstract**

Generating responses that are consistent with the dialogue context is one of the central challenges in building engaging conversational agents. We demonstrate that neural conversation models can be geared towards generating consistent responses by maintaining certain features related to topics and personas throughout the conversation. Past work has required external supervision that exploits features such as user identities that are often unavailable. In our approach, topic and persona feature extractors are trained using a self-supervised discriminative training scheme that utilizes the natural structure of dialogue data. We further adopt a feature disentangling loss which, paired with controllable response generation techniques, allows us to promote or demote certain learned topics and persona features. Evaluation results demonstrate the model's ability to capture meaningful topics and persona features. The incorporation of the learned features brings significant improvement in terms of the quality of generated responses on two datasets.

#### 1 Introduction

The notion of dialogue consistency is attracting growing interest in multi-turn neural response generation research (Li et al., 2016b; Luan et al., 2016; Zhang et al., 2018a). When interacting with an open-domain neural conversation agent, users may expect the agent to develop the dialogue with consistent information to mitigate confusion and improve engagement. We consider two aspects of consistency: topic consistency and persona consistency. In our definition, topic consistency is specific to a multi-turn dialogue session. Topic consistency reflects the model's ability to maintain dialogue topics/themes such as sport, movie or music without becoming sidetracked. On the other hand, persona consistency is specific to a

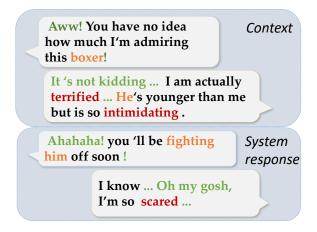


Figure 1: Task illustration: generating responses that are consistent with dialogue history in persona, tone and topic (from our system, better in color).

speaker. Persona consistency envisions the agent as human-like, endowed with a relatively invariant personality (e.g., be like Harry Potter), style of engagement (e.g., enthusiasm and casualness) or personal profile (e.g., place of residence). A case of violation of persona consistency may be the agent conflicts itself by sometimes saying it lives in London, sometimes saying it lives in Paris. Generating appropriate responses with these characteristics is a major challenge (Figure 1). (Li et al., 2016b; Luan et al., 2017) and (Al-Rfou et al., 2016) use persona embeddings as additional input to train end-to-end conversational agents. Obtaining accurate persona embeddings as in (Li et al., 2016b) however requires many thousands of utterances per persona, and targeted test personas may not always be found in the training data. End-to-end systems are often trained from social media data in which only a small spectrum of personas (casual speakers) is well represented, while professional roles (e.g. customer service) are underrepresented, thus limiting deployment. Typically, moreover, the objective is to maintain consistency of both persona and topic throughout the dialogue, rather than to inject

specific personas/topics into responses, making it crucial to learn and leverage both in a data-efficient and unsupervised manner.

In this paper, we present a self-supervised approach (a subdomain of unsupervised methods) that exploits the natural structures of conversational data to efficiently learn and leverage topic and persona features. Our proposal features: 1) A discriminative feature extraction mechanism to capture dialogue topics and personas in a self-supervised manner, without requiring specification of speaker identity, thus i) allowing massive unlabeled data to be utilized and ii) protecting sensitive personallyidentifiable ID information. 2) Use of binary features and a disentangling loss to improve interpretability of learned features. This affords flexibility to activate or deactivate specific features when generating responses. 3) Leveraging a controllable text generation mechanism to force generated responses to adhere to high-level features such as topic and persona encoded in the controlling signal.

## 2 Related Work

Self-supervised learning Self-supervised learning as a subdomain of unsupervised learning, has been applied to representation learning for image, video and audio (Denton and Vighnesh, 2017; Doersch et al., 2015). Borrowing definitions from other domains, self-supervised approaches in NLP make use of non-textual signals that intrinsically correlate with the text to supervise text feature learning (Denton and Vighnesh, 2017). However, to the best of the authors' knowledge, self-supervision has yet to be applied to conversational agents.

Persona-aware response generation (Welleck et al., 2018) suggested a natural language inference approach to improve persona consistency, but this requires additional labels. (Zhang et al., 2018a; Qian et al., 2018) use explicit personal profiles as side information to guide response generation. Such information, however, is often available. Other work proposes injecting either emotion (Zhou et al., 2018) or functional control (Ke et al., 2018) into dialogue generation. As in (Li et al., 2016b), learning to leverage controlling signals in order to bias generation may require significant amounts of labeled data.

**Topic-aware response generation** Leveraging topic modeling in response generation has been

explored by several prior works (Xing et al., 2017; Wang et al., 2017; Wu et al., 2018). Our approach differs from these methods in that our topics are learned by discriminating the source of utterance pairs. Our method employs a neural sentence encoder to capture richer features than the bag-of-words features of conventional topic models.

Interpretable and controllable generation Controllable text generation (Hu et al., 2017) has been employed in text style transfer and many other tasks (Ficler and Goldberg, 2017; Asghar et al., 2018; Ghosh et al., 2017; Dong et al., 2017). This helps disentangle high-level style information from contextual information such that the style information can be independently manipulated to produce text with different styles. Related to our work, (Zhao et al., 2018) uses discrete latent actions to learn an interpretable representation for task-oriented dialogue systems.

Generic Pretraining methods Generic pretraining methods (e.g. BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2018), DialoGPT(Zhang et al., 2019)) for representation learning/language generation has recently gained much attention. At a very high level, BERT and our method share the idea of "unsupervisedly-trained feature extractors", but the goals and methods are quite different: BERT is trained using masked language model and next sentence prediction. Our self-supervised feature extractor, instead, is trained by distinguishing paired utterances from same/different speakers/dialogues. Thus, unlike BERT, which abstracts generic features, our self-supervised objective abstracts the discriminative features that differentiate two speakers in a dialogue, thus emphasizing sentence styles (e.g. tone, persona) rather than sentence content. Although leveraging pretrained BERT to initialize our feature extractor might conceivably improve the results presented in this proof-of-concept paper, modifying our current RNN-based framework to a transformer-based one is non-trivial and must be left to future work.

## 3 Method

The proposed approach uses additional unsupervisedly learned features to generate response utterances that reflect these features. We elaborate two major components of the proposed approach: feature extractors trained to extract topic/persona features from each utterance; and a response gen-

*erator* that takes the extracted features as input to generate responses accordingly.

#### 3.1 Problem statement

Let  $d^{(i)} \triangleq [u_1^{(i)}, u_2^{(i)}, \cdots, u_T^{(i)}]$  denote the i-th dialogue in dataset  $\mathcal{D}$ , where  $u_j^{(i)}$  is the j-th utterance and T is the number of turns in this dialogue. We assume that each dialogue  $d_i$  only consists of the utterances between two speakers, interleaving with each other. Suppose the first K (K < T) turns of each dialogue are revealed, our aim is to generate the remaining T - K turns of the dialogue that are consistent with the observed context.

#### 3.2 Discriminative feature extraction

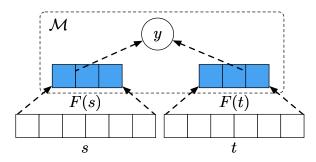


Figure 2: Feature extractor design. s and t are two randomly shuffled sentences. An extractor network  $F(\cdot)$  (F can be either  $\mathcal{T}$  or  $\mathcal{P}$ ) encodes both of them to yield features F(s) and F(t), which are then used to predict label y.  $\mathcal{M}$  represents matching function/network.

Inspired by (Denton and Vighnesh, 2017), we adopt a self-supervised discriminative training scheme, where we design a neural model that includes a feature extraction layer as illustrated in Figure 2 and formulate a discriminative task to train the model. When the training is done, the feature extraction layer yields relevant features for the associated task. In this section, we introduce two discriminative tasks to capture two types of sentence features, respectively: 1) *topic features*(or, session-specific features,  $\mathcal{T}$ ) that characterize conversation topics. 2) *persona features* (or, speaker-specific features,  $\mathcal{P}$ ) that reflect speaker characteristics.

**Topic/session-specific feature extractor** In building a topic/session-specific feature extractor with self-supervision, we rely on the assumption that utterances from the same conversation session are likely to share similar topics. Thus, we formulate a surrogate task to determine if two random sentences s and t from  $\mathcal{D}$  belong to the same dialogue session. Specifically, when they

come from the same dialogue, *i.e.*  $s, t \in d_i$ , we assign 1 to target y and 0 otherwise. We optimize the cross-entropy objective:

$$\mathcal{L}_{ exttt{xent}} = rac{1}{N} \sum_{i=1}^{N} [y_i \log \mathcal{M}(\mathcal{T}(s_i), \mathcal{T}(t_i)) + (1 - y_i) \log [1 - \mathcal{M}(\mathcal{T}(s_i), \mathcal{T}(t_i))]],$$

where  $\mathcal{T}(\cdot)$  denotes the topic feature extractor (shared among all sentences), and  $\mathcal{M}(\cdot,\cdot)$  represents a matching network, detecting whether the two feature vectors  $\mathcal{T}(s_i)$  and  $\mathcal{T}(t_i)$  belong to the same dialogue session. We use a convolutional neural network (CNN) followed by a non-linear mapping for  $\mathcal{T}(\cdot)$  to produce an L-dimensional vector  $\mathcal{T}(x)$ . We explore two options for the  $\mathcal{T}(\cdot)$ : 1) We employ a sigmoid function to produce a soft-binary representation, i.e.  $\mathcal{T}(x) \in (0,1)^L$ . This nonnegative bounded representation lends itself well to interpretation and control of each component of  $\mathcal{T}(x)$ . For the matching function  $\mathcal{M}(\cdot)$ , we apply a sigmoid function to the inner-product of two feature vectors, i.e.  $\mathcal{M}(\mathcal{T}(s), \mathcal{T}(t)) = \sigma(\mathcal{T}(s))$  $\mathcal{T}(t)/\tau$ ), where  $\tau$  is a temperature hyperparameter. A feature being close to 1 (close to 0) indicates being activated (deactivated). Intuitively, if the same feature  $(\mathcal{T}(s)_m \text{ and } \mathcal{T}(t)_m)$  from both sentences are activated, this feature will contribute more to the final prediction that the pair is from the same dialogue. Presumably, this feature represents a certain dialogue topic that is shared by the sentence pair, based on the assumption that utterances from the same conversation session are much likely to overlap in topic, compared to utterances from a different conversation session. 2) Alternatively, we compute a hard-binary representation by taking only 0 and 1, i.e.  $\mathcal{T}(x) \in \{0,1\}^L$ , which lends itself well to a discrete control and crisp interpretation. A straight-through (ST) estimator is used for gradient calculation (Bengio et al., 2013). Suppose the binary feature F is rounded from a probability vector p, the ST estimator back-propagates through the discrete decision by approximating the gradient  $\partial F/\partial p$  as 1. We empirically found that setting  $\mathcal{M}$ to use the inner product of F(s) and F(t) results in poor performance. We presume that the inner product between two binary vectors, which can only take integers from [-L, L], limits the representation power of the model. Thus we concatenate F(s)and F(t) and passing it through a multi-layer perceptron (MLP) to predict matching label y. Note that we assume each conversation session can contain multiple topics. The topic feature extractor essentially explicitly captures ooccurrence of n-gram appearing in one conversation session, and can use this information to hint the model to relate to learned topics.

Persona/speaker-specific feature extractor Here we consider extracting persona/speakerspecific features in a broader sense of current speaker's status related to emotion (Zhou et al., 2018), personality, tone and function control (Ke et al., 2018). Note that we are only interested in maintaining consistency of emerging persona, rather than characterizing a full spectrum of persona features. The only difference between the topic  $(\mathcal{T})$  and persona  $(\mathcal{P})$  feature extractors is how the positive and negative sample pairs for training are created. In the persona feature extractor, the positive pairs (y = 1) or negative pairs (y = 0) are the utterances from the same or different speaker within a dialogue to keep the topic the same. Ideally, two speakers in a dialogue are discussing the same topic. Thus, the model is encouraged to learn features that capture their different personas rather than different topics. Note that unlike (Li et al., 2016b), where separate embedding vectors are allocated for each speaker, in our proposed method the utterances by one speaker can have different feature vectors as the manifestations of the underlying persona embedding in a different context. We believe that our approach that abstracts the utterance feature is more data-efficient than (Li et al., 2016b) which abstracts the persona feature, because 1) The utterance features allow borrowing information from a wider range of speakers, as long as they share traits with the current speaker in the current context. For example, when talking about sports, utterance-based feature learning can benefit from all sports lovers' utterances, even if their overall persona features are not close. 2) For an unseen speaker at testing time, unlike a persona feature, utterance features can still be inferred easily even if there are no training instances for the new speaker.

**Disentangling loss** To make the learned features more *interpretable*, we further employ a *decorrelation* (DeCorr) loss inspired by (Cogswell et al., 2015), who introduced a DeCov loss to regularize deep neural networks. Specifically, we add an additional term in the objective function when training

the topic and persona feature extractors:

$$\begin{split} \mathcal{L}_{\text{DeCorr}} &= \frac{1}{2} (||M||_F^2 - ||\text{diag}(M)||_F^2) \\ M_{jk} &= \frac{\sum_i (F(s_i)_j - \mu_j) (F(s_i)_k - \mu_k)}{\sqrt{\sum_i (F(s_i)_j - \mu_j)^2 \sum_i (F(s_i)_k - \mu_k)^2}}, \end{split}$$

where  $||\cdot||_F$  represents the matrix Frobenius norm, and the  $\mathtt{diag}(\cdot)$  operator represents diagonalization of a matrix. F is the feature extractor, and can be either  $\mathcal{T}$  or  $\mathcal{P}$ . M is the correlation matrix of F, computed from the current batch of data (i runs through a mini-batch).  $\mu$  represents the mean feature over mini-batch, i.e.,  $\mu_j = 1/N \sum_{i=1}^N F(s_i)_j$ . Achieving a reasonable estimation of the correlation matrix requires a relatively large mini-batch size. The resulting final objective for the discriminator is:

$$\mathcal{L}_{D} = \mathcal{L}_{\text{xent}} + \lambda \mathcal{L}_{\text{DeCorr}},$$

where  $\lambda$  is a balancing hyperparameter.

Utterance pair construction One issue in constructing the positive/negative pair for the feature extractors is that the number of positive/negative pairs need to be balanced to achieve a robust empirical result. Moreover, when constructing positive pairs (y = 1), if the s and t are adjacent or close to each other in a dialogue, we might end up capturing adjacency pairs (Sacks and Schegloff, 1973) rather than conversational topics. For example,  $s = How \ are \ you?$  and t = Fine. How are you?. The captured similarity in the feature space of this pair is just a contextual appropriateness rather than topic/persona consistency. To alleviate this, we collect only those pairs that are more than 4 turns away from each other for positive pairs. We also note that persona features may influence the topic feature extractor because persona features can be weak signals for predicting whether two sentences are from the same dialogue. One remedy is to select utterances from different speakers within a dialogue session when constructing positive pairs for the topic extractor to minimize the influence of persona features. However, this remedy can result in fewer positive pairs. Empirically, the topic extractor works well even without this remedy, presumably because other signals for topic extraction may overwhelm the weak signal from persona features. We provide some additional comments in the Appendix.

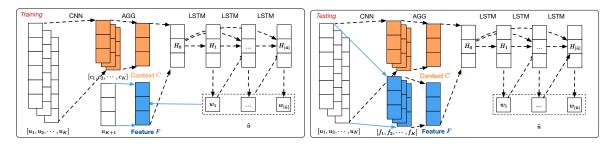


Figure 3: Controllable generation scheme (better in color). The feature extractors are represented as solid blue arrows. Left: For training, contextual sources  $[u_1, \cdots, u_K]$  are encoded, and aggregated to a context vector C, meanwhile the target  $u_{K+1}$  is abstracted by feature extractor as feature vector F. The decoder then (controllably) generates a response  $\tilde{u}$  based on C and F. Right: For testing, the feature F is obtained by aggregating the feature vectors for each source sentence.

## 3.3 Generator design

**Training a response generator** The conditional multi-turn generator that produces neural responses given previous K-turn source sentences is shown in Figure 3, which is conceptually related to (Serban et al., 2016). During training time (Figure 3 left panel), each source sentence is first encoded by a 3layer CNN encoder, which shares the same architecture as the feature extractor, followed by a context aggregator  $(AGG_C)$  layer that summarizes all sentence embedding vectors  $[c_1, c_2, \cdots, c_K]$  into one single context vector C with the same dimension as  $c_i$ . In this paper, the  $AGG_C$  layer is designed as first concatenating  $[c_1, c_2, \cdots, c_K]$  and applying a fully-connected layer to map the resulting vector to C. On the other hand, the target sentence is processed by the feature extractors to produce feature vector(s) F (see "Discriminative feature extraction"). The feature extractors are fixed in the response generator since we observed that simultaneously optimizing the feature extractor and the decoder under generator loss leads to suboptimal empirical results, as the extractor will forget the learned feature from self-supervised learning loss  $\mathcal{L}_{\text{xent}}$ . The context vector C and feature vector(s) F are fed into an MLP to generate a fixed-length initial hidden variable  $H_0$ . This is followed by a series of long short-term memory (LSTM) units of the decoder, where  $H_0$  is employed as input in each time-step.

Controllable objective during training Our generator loss incorporates two components. The first is the vanilla teacher-forcing (Williams and Zipser, 1989) MLE loss  $\mathcal{L}_{\text{MLE}}$ . The second is a cycle consistency loss  $\mathcal{L}_{\text{cycl}}$ , introduced by (Hu et al., 2017) to admit an additional controlling ability of feature vector in the generation. Intuitively,

it encourages the generated response to possess the same features as the input features F. Specifically, consider a response  $\tilde{u} = [w_1, w_2, \cdots, w_{|\tilde{u}|}]$  greedily generated by conditioning on previously generated tokens. The  $\mathcal{L}_{\text{cvcl}}$  is simply the Euclidean distance between input feature vectors F and  $F(\tilde{u})$ , i.e.  $\mathcal{L}_{\text{cycl}} = ||F - F(\tilde{u})||^2$ . In the case of binary features,  $\mathcal{L}_{\text{cycl}} = ||F - P(\tilde{u})||^2$  where  $P(\cdot)$  is the network output before rounding to binary values. Note that the generated tokens  $[w_1, w_2, \cdots, w_{|\tilde{u}|}]$ involve an argmax operation and are not directly differentiable, preventing the gradient signals from back-propagating to the encoder and decoder. Common remedies for this includes Gumbel-softmax (GS) (Gumbel and Lieblein, 1954), policy gradient (PG) (Yu et al., 2017) and soft-argmax (SA) (Zhang et al., 2017a). Unfortunately, GS and PG suffer from high variances of gradient estimation while SA suffers from a dilemma between gradient vanishing and inaccurate gradient. To alleviate the issue with SA, we consider an approach called Straight-Through LSTM unit (ST-LSTM), which uses ST estimation (Bengio et al., 2013) to achieve a biased but smooth gradient signal while maintaining the forward computation exact via a temperature parameter  $\tau$ . The details are provided in the Appendix. In the experiment, we applied the slope-annealing trick (Chung et al., 2016), and set  $\tau = 0.01$ . The final training objective for generation is:

$$\mathcal{L}_{\text{G}} = \mathcal{L}_{\text{MLE}} + \eta \mathcal{L}_{ ext{cycl}}$$

**Testing time** At test time, as shown in Figure 3 (right panel), the feature vectors from the source sentences  $[u_1, \cdots, u_K]$  are first collected by applying feature extractors  $F(\cdot)$ . We denote the feature vectors for the source sentences as  $[f_1, \cdots, f_K]$ . We apply a feature aggregator

 $AGG_F$  layer to estimate the output feature vector F, which is further fed into the LSTM-RNN for generation. Different from the context aggregator  $AGG_C$  layer, we consider a weightedsum aggregation function for the feature  $AGG_F$ layer, i.e.,  $F = \sum_{k=1}^{K} w_k f_k$ , s.t.  $\sum_{k=1}^{K} w_k = 1$ , where  $w_k, k = [1, 2, \dots, K]$  are linear interpolation weights learned during training time, where a Euclidean distance between predicted features and target features is optimized, *i.e.* argmin<sub>w</sub> $\mathcal{L}_p =$  $||f_{k+1} - \sum_{k=1}^{K} w_k f_k||^2$ . For persona features, we only use the source sentences of the current speaker, thus all  $w_k$  where mod(k, 2) = mod(K, 2) are set to zero. This weighted averaging can be interpreted as an attention over utterances. The additional input features do not aim to "fix" the dialog under certain topics/persona, but to "facilitate" the model to leverage dialogue history to generate new utterance in a totally data-driven manner. We note that more complicated attention mechanisms can further improve the model; however, we leave these for future work, since this paper focuses on the self-supervised learning for dialogue features rather than improving the multi-turn S2S structure in general.

## 4 Experimental Setups

We evaluate the proposed methods on two datasets. All experiments are conducted using single Nvidia Tesla V100 GPU. The source code will be released.

#### 4.1 Data collection

We consider two datasets. For both, we use a (80%, 10%, 10%) split for training, validation and test respectively. **Twitter data** Training data was extracted from the Twitter FireHose covering a five-year period from 2012 through 2016.<sup>2</sup> From this set, we collected total 6,658,385 8-turn (4-turn context) dialogues where two participants chatted with each other. Comments regarding the number of context turns are provided in the Appendix. **MetaLWOz data** Orthogonal to the twitter dataset, the MetaLWOz dataset (Lee et al., 2019) consists of 40,389 task-oriented conversational interaction between two speakers regarding 51 domains and 242 tasks, collected by crowd-sourcing where one

crowd worker simulates a user and another simulates a bot. Each dialogue has 11 turns. We evaluate our model on this dataset for two reasons: 1)the number of turns is high, enabling evaluating the model's capability of capturing the long-range dependency; 2)domain labels are available, enabling visualization of the unsupervised learned features by each label.

### 4.2 System specifications

The model training and hyperparameter details are provided in the Appendix. For evaluation, we consider three variants of our COnsistent CONversation (CoCon) models: CoCon-T: CoCon model with topic-consistency; CoCon-TP: Co-Con model with topic-consistency and personaconsistency; CoCon-TP-bin: CoCon model using binary features with topic-consistency and personaconsistency. Note CoCon-P is not included in our experiment, as the comparison between CoCon-T and CoCon-TP can be used to assess the effectiveness of adding the persona feature (-P). We compare our models with two baselines: a vanilla sequence-to-sequence model (S2S) and persona model (Persona) (Li et al., 2016b). Note that our method does not conflict, and can be stacked with any of the hierarchical approaches (Serban et al., 2017; Zhao et al., 2018), thus these approaches are not included in our comparison. The proposed algorithm is most comparable with (Li et al., 2016b) as they are mutually exclusive. We implement the persona model by reusing the encoder and decoder architecture. For the Twitter dataset, we map all users with fewer than 88 utterances as unknown (86% of the total training samples) and in the test set (for all compared methods) we eliminate conversation sessions with unknown users. This yields 50k total users. Note that this setting presents a systemic advantage to the **Persona** baseline. We use the same number of feature dimensions for all systems. To ensure that all modules training have converged, we use a patience parameter of 10 epochs to wait before early stopping until no progress on validation loss.

### 5 Results

**Self-supervised feature learning** We used equal numbers of positive/negative examples to train each feature extractor. For the Twitter data, the resulting accuracy for the topic and persona feature extractors is around 0.75 and 0.60 (for both

<sup>&</sup>lt;sup>1</sup>Other possibilities for the  $AGG_F$  layer exist, such as mean, max or concatenation (as in  $AGG_C$ ). We choose weighted-sum for the  $AGG_F$  layer due to its superior empirical performance compared to alternatives.

<sup>&</sup>lt;sup>2</sup>Deleted tweets and closed accounts were removed.

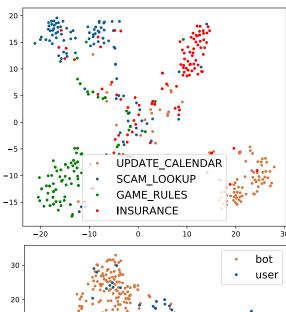
Ü		3gram	4gram
android;	the iphone;	a new phone;	is on my phone; you
ios;	the app; my	my phone is;	can use the; send it
apps	ipad	are you us-	to the
		ing	
striker;	champions	in the	one of the best; best
arsenal;	league ;	league; in	of the season; the
madrid	best player;	the playoffs;	team in the;
	the spurs;	a good game	
episodes;	of thrones;	i watched it;	one of my favorite;
film;	that movie;	to watch it;	have you seen the; i
netflix	the ending	the first one	want to see
1 gram	2gram	3gram	4gram
what;	can u; do u;	how do i; is	thank you so much; i
tlavr.	4		
thx;	where is	there a; what	need to get; i want
wheres	where is	there a; what is the	need to get; i want to see
	lol yea; yea	,	•
wheres		is the	to see
wheres yea;	lol yea; yea	is the that s so; i	to see i would love to; look-
wheres yea; great;	lol yea; yea	is the that s so; i will do; yea i	to see i would love to; looking forward to it;
wheres yea; great; sure	lol yea; yea i; hell yea	is the that s so; i will do; yea i m	to see i would love to; looking forward to it; sounds like a plan
	striker; arsenal; madrid episodes; film; netflix lgram what;	android; the iphone; ios; the app; my apps ipad  striker; champions arsenal; league; madrid best player; the spurs; episodes; of thrones; film; that movie; netflix the ending  lgram 2gram what; can u; do u;	android; the iphone; a new phone; ios; the app; my my phone is; are you using  striker; champions in the arsenal; league; league; in madrid best player; the spurs; a good game episodes; of thrones; i watched it; netflix the ending the first one  1 gram 2 gram 3 gram  what; can u; do u; how do i; is

Table 1: Representative n-grams in topic (T) and persona (P) features.

continuous and binary features), respectively. For the MetaLWOz dataset, the discriminator accuracy for the persona and topic feature extractors is 0.85 and 0.67, respectively. With the disentangling loss ( $\lambda=0.01$ ), the correlation between features drops from 0.25 to 0.16. Representative n-grams of some learned features from the Twitter dataset are shown in Table 1. To calculate the feature vector for a specific n-gram, we average over the feature vectors of test sentences that contain that n-gram. We then select the top-ranked n-grams with occurrences greater than 200 for each feature. We further visualized topic features

Context:	A:tomorrows game just became that much bigger . must win .
	B: unbeliebable . mourinho has never lost at home in the
	league as chelsea coach and against the worst team in the
	league he loses
	A:yup, would be insane if they could somehow avoid relega-
	tion now.
	B:would be amazing.we played against them at home 3 weeks
	ago but barely beat them . they are always up for it against the
	big teams
Human:	A:yeah, said they played against top half of table much better
	then bottom half.
S2S:	A:i think they 'll be fine
Persona:	A:i think they 'll be in the same position .
CoCon-	A:i think they play well . i 'm a big fan of the manager . they
T:	've <b>played</b> well in the same <b>league</b> as well.
CoCon-	A:yeah, they played well with the top three teams, but i'd
TP:	say it was . not sure what to expect .
Context:	A:lucky you mate . have a super fine weekend buddy
	B: i m gonna be knackered next week should be a good week-
	end though gonna get loads a pics
	A:can n't fault you mate nice mid night start tuesday lol
	B:gonna be at the show saturday then home sunday back to
	work monday . where you off to tomorrow ?
Human:	A:one out of cedar farm to march then not sure mate either
	ipswich or a hiams to march.
S2S:	A:not sure . i 'm off for a few days
Persona:	A:i 'm off friday night . you going to the game ?
CoCon-	A:i know . i am going to manchester on the weekend . what
T:	about you ?
CoCon-	A:i 'm going to the beach mate . i 'm going for a couple days
TP:	off . good weekend pal

Table 2: Sampled generation on Twitter.



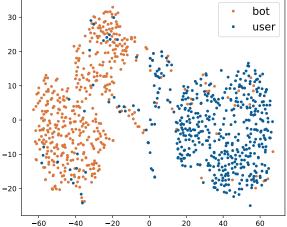


Figure 4: t-SNE embedding of topic (left) and persona (right) features for different tasks (left) and roles (right).

Context	A:done with the <b>game</b>
	B:what are you talking about
	A:you heard me
	B:if you 're not there today i 'm going to be mad
Human:	A:lol ok i aint going
	B:are you trying to be a bod mon or something?
	A:oh you that 's exactly how i 'm tryna be aha
	B:so tell me why you aren't going?you just say you're not, talk
	about it
Plain:	A:i'm on the vacation. i'm not going to play that game
	B:you don't really mean this, right? you're n't going to do that
	A:i aint, i told you.
	B:well you can n't be a person like that, you can n't be a loser
Sport:	A:i'm watching the school <b>football</b> . that's enough for now
on	B:that's why you're a loser
	A:lol no i'm going to be the best <b>player</b> in the <b>league</b> .
	B:oh well you're not going to win man you are terrible at all this
	time.

Table 3: Examples of feature manipulations. *Plain* denotes without manipulation. X on denotes feature X is activated (set to 1) when generating the response.

on both datasets using t-SNE embedding (Maaten and Hinton, 2008). For the MetaLWOz dataset, Figure 4 illustrates the learned topic and persona feature embeddings on the test data. Without any label information, the learned topic and persona features separate well. For the twitter dataset,

we observed that the persona features of the utterances from different time zones form clusters, indicating the features learned by our method can reflect differences in societal groups (Appendix, Figure 6).

Sampled response generation We evaluate our approaches by generating a response given 4 previous turns. Some sampled results are shown in Table 2. We observed that the CoCon-T and CoCon-TP can produce informative responses which is in general more consistent with the given context compared to baselines. For CoCon-TP, beyond being context-aware, the responses seem to be personaware, *i.e.*, mimicking the tone and personal wording preferences like *mate*, *oh my gosh*, *haha*, and *ain't* and other words associated with them.

Feature manipulation We further manipulate features that seem to be associated with certain topics. The results are shown in Table 3 (additional results are in Appendix). We generate next 4 turns consecutively conditioned on 4 source sentences, including previously generated utterances as source context. This feature manipulation is based on hardbinary features, achieved by toggling a specific feature to be 1 to activate it. the corresponding feature  $\tilde{F}$  extracted from the generated response is also active. Presumably, the model has learned that, based on the context, it is *unnatural* to toggle the feature. This hypothesis needs further experimental verification. For the MetaLWOz dataset, the context is given as 4 turns of dialogues and the task is to generate all remaining 7 turns. We found that the S2S model tends to generate looping responses like thanks and you're welcome and is generally less informative. However, our proposed CoCon-TP approach can generate reasonably responses (e.g. ticket booking details) by unsupervisedly capturing the topics of the context and role of each turn. (Additional sampled responses for S2S and CoCon-TP are provided in Appendix Table 8).

Automatic evaluations The dialogue evaluation is challenging (Liu et al., 2016). Thus, we seek to compare systems over many automatic metrics that cover different aspects. In our quantitative evaluations, we test both *relevance* and *diversity* metrics. For relevance, we adopt BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), NIST (Doddington, 2002) and three embeddingbased metrics Greedy, Average, Extreme following (Serban et al., 2017; Rus and Lintean, 2012;

Mitchell and Lapata, 2008; Forgues et al., 2014). To evaluate diversity, we follow (Li et al., 2016a) to use **Dist-1** and **Dist-2**, characterized by the proportion between the number of unique n-grams and total number of n-grams of tested sentence. We also include the **Entropy** (Ent-n) metric (Zhang et al., 2018b), which does not depend on the size of test data. The results of automatic evaluations are shown in Table 4 (Twitter) and Table 5 (MetaLWOz). For both dataset, the CoCon-TP model achieves the best relevance score, while the CoCon-TP-bin outperforms other methods in diversity.

Ablation study of the auxiliary losses We observe that, when  $\lambda = 0$ , *i.e.*, without disentangling loss, the learned features exhibit heavy colinearity. This may weaken the interpretability of each feature, and is less efficient in feature vector utilization, as similar "topics" will occupy many feature slots. Further details are provided in the Appendix. With the additional controllable generation objective  $\mathcal{L}_{CVCl}$ , we are able to better control the flipping of each feature. As shown in Figure 5, increasing  $\eta$  leads to a fast decrease of  $\mathcal{L}_{\texttt{cycl}}$  (indicating a better controlling power), however it may come at the cost of generation quality. We select the  $\eta = 0.01$  to balance the trading-off between both aspects. We observe the success rate of feature toggling is about 17% percent (based on 2000 test cases), meaning that around 17% cases where we activate an input feature F.

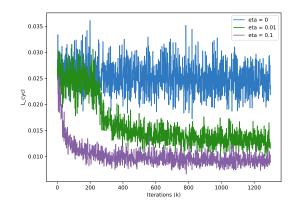


Figure 5:  $\mathcal{L}_{ t cycl}$  decreases faster when  $\eta$  is larger.

Human evaluations We evaluated 500 randomly sampled test sources from the Twitter dataset by crowd-sourcing. Systems were paired and each pair of system outputs was randomly presented to 4 judges, who ranked them for topic consistency, persona consistency, informativeness and relevance

	Models		Relevance						Diversity		
	Models	BLEU	METEOR	NIST	Greedy	Average	Extreme	Dist-1	Dist-2	Ent-4	
Ours	CoCon-T	3.01	0.061	1.022	1.968	0.675	0.321	0.008	0.065	9.71	
Ours	CoCon-TP	3.31	0.064	1.135	2.048	0.683	0.342	0.008	0.081	10.46	
	CoCon-TP-bin	3.04	0.063	1.061	2.025	0.677	0.331	0.009	0.100	10.59	
	S2S	2.83	0.056	0.945	1.855	0.640	0.307	0.004	0.023	7.51	
Pe	ersona model	2.96	0.059	1.014	1.931	0.658	0.319	0.005	0.028	7.96	
	Human	-	-	-	-	-	-	0.078	0.473	11.75	

Table 4: Quantitative evaluation for twitter dataset

	Models	Relevance						Diversity		
	Models	BLEU	METEOR	NIST	Greedy	Average	Extreme	Dist-1	Dist-2	Ent-4
Ours	CoCon-T	5.6	0.076	1.421	2.105	0.565	0.358	0.025	0.142	8.899
Ours	CoCon-TP	5.8	0.077	1.459	2.175	0.575	0.365	0.028	0.16	8.983
	CoCon-TP-bin	4.6	0.074	1.280	2.094	0.559	0.341	0.027	0.19	9.767
	S2S	3.9	0.066	1.045	2.021	0.529	0.328	0.017	0.16	8.293
Pe	ersona model	4.4	0.073	1.134	2.042	0.543	0.319	0.021	0.177	8.603
	Human	-	-	-	-	-	-	0.092	0.462	10.281

Table 5: Quantitative evaluation for MetaLWOz dataset

Topic Consistency (human judges prefe	Persona Consistency (human judges preferred)					
Our Method   Neutral   Comparison		Our Method		Neutral Compar		arison
CoCon-TP         45.20%         22.30%         32.50%           CoCon-TP         40.05%         23.10%         36.85%	seq2seq persona	CoCon-TP CoCon-TP	40.95% 35.65%	29.85% 34.10%	29.20% 30.25%	seq2seq persona
CoCon-TP 21.50%   26.85%   <b>51.65</b> %	human	CoCon-TP	21.35%	33.35%	45.30%	human

Table 6: Results of **Human Evaluation** for topical and persona consistency, showing preferences (%) for our model (CoCon-TP) vis-a-vis baseline or other comparison systems. Distributions are skewed towards CoCon-TP, except when compared with human outputs. Numbers in bold indicate the most preferred systems. For simplicity, the 5-point Likert scale is collapsed to a 3-point scale. See the Appendix for further details.

using a 5-point Likert scale. The judges are given 4-turn dialogue history in order to assess the consistency and coherence of the generated response. Overall judges' preferences for the topic consistency, persona consistency, given as a percentage of total judgments are shown in Table 6. A strong preference is observed for CoCon-TP over the other systems. We also evaluated for relevance and informativeness, with CoCon-TP showing similar preference gains. Further details, including Krippendorff's alpha for inter-rater consistency and the human evaluation template used, are provided in Appendix. We omit the human evaluation of MetaLWOz dataset for budget reason in a hope that the twitter human evaluation can adequately assess the compared systems.

## 6 Conclusion

We present a self-supervised feature learning framework to abstract high-level latent representations of topic and persona information underlying the dialogue context and leverage these representations to generate more consistent dialogue in a controllable manner. Compared with previous works, it delivers superior empirical results and is more data-efficient. In future work, leveraging pretrained model to initialize our feature extractor could potentially improve the results. Our focus here, however, has been on presenting a general framework that can stand as a building block for multiple architectures. Combining and aligning supervised and unsupervised features are should further improve feature learning and interpretability. We expect our method can also facilitate style transfer and long-form generation to improve consistency.

#### References

- Rami Al-Rfou, Marc Pickett, Javier Snaider, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2016. Conversational contextual cues: The case of personalization and history for response ranking. *arXiv*.
- Nabiha Asghar, Pascal Poupart, Jesse Hoey, Xin Jiang, and Lili Mou. 2018. Affective neural response generation. In *ECIR*. Springer.
- Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv*.
- Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. 2016. Hierarchical multiscale recurrent neural networks. *arXiv*.
- Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. 2015. Reducing overfitting in deep networks by decorrelating representations. *arXiv*.
- Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *workshop on statistical machine translation*.
- Emily L Denton and Birodkar Vighnesh. 2017. Unsupervised learning of disentangled representations from video. In *NIPS*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.
- George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc.
- Carl Doersch, Abhinav Gupta, and Alexei A Efros. 2015. Unsupervised visual representation learning by context prediction. In *ICCV*.
- Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *EACL*.
- Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. *EMNLP*.
- Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. 2014. Bootstrapping dialog systems with word embeddings. In NIPS, modern ML and NLP workshop.
- Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. 2017. Affect-LM: A neural language model for customizable affective text generation. In *ACL*.
- Emil Julius Gumbel and Julius Lieblein. 1954. *Statistical theory of extreme values and some practical applications: a series of lectures.* US Government Printing Office Washington.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *ICML*.
- Pei Ke, Jian Guan, Minlie Huang, and Xiaoyan Zhu. 2018. Generating informative responses with controlled sentence function. In *ACL*.

- Sungjin Lee, Hannes Schulz, Adam Atkinson, Jianfeng Gao,
   Kaheer Suleman, Layla El Asri, Mahmoud Adada, Minlie
   Huang, Shikhar Sharma, Wendy Tay, and Xiujun Li. 2019.
   Multi-domain task-completion dialog challenge. In *Dialog System Technology Challenges (DSTC) AAAI Workshop*.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In NAACL.
- Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. A persona-based neural conversation model. In ACL.
- Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike 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. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.
- Yi Luan, Chris Brockett, Bill Dolan, Jianfeng Gao, and Michel Galley. 2017. Multi-task learning for speaker-role adaptation in neural conversation models. In *Proceedings of* the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers).
- Yi Luan, Yangfeng Ji, Hannaneh Hajishirzi, and Boyang Li. 2016. Multiplicative representations for unsupervised semantic role induction. In *ACL*.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9.
- Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *ACL*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In ACL.
- Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Assigning personality/identity to a chatting machine for coherent conversation generation. In *IJCAI*.
- A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. 2018. Language models are unsupervised multitask learners. Technical report, OpenAI.
- Vasile Rus and Mihai Lintean. 2012. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*.
- Harvey Sacks and Emanuel A. Schegloff. 1973. Opening up closings. *Semiotica*, 8(4):289327.
- Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Hierarchical neural network generative models for movie dialogues. In AAAI.
- Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In AAAI.
- Di Wang, Nebojsa Jojic, Chris Brockett, and Eric Nyberg. 2017. Steering output style and topic in neural response generation. In *EMNLP*.

- Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2018. Dialogue natural language inference. *arXiv*.
- Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.
- Yu Wu, Zhoujun Li, Wei Wu, and Ming Zhou. 2018. Response selection with topic clues for retrieval-based chatbots. *Neu-rocomputing*.
- Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *AAAI*.
- Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: sequence generative adversarial nets with policy gradient. In *AAAI*.
- Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018a. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv*.
- Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018b. Generating informative and diverse conversational responses via adversarial information maximization. In *NeurIPS*.
- Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017a. Adversarial feature matching for text generation. In *ICML*.
- Yizhe Zhang, Dinghan Shen, Guoyin Wang, Zhe Gan, Ricardo Henao, and Lawrence Carin. 2017b. Deconvolutional paragraph representation learning. In NIPS.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation.
- Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. 2018. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. *arXiv*.
- Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In AAAI.

## **Appendix for Consistent Dialogue Generation with Self-supervised Feature Learning**

## A Straight-through LSTM (ST-LSTM)

In the forward calculation, the t-th ST-LSTM unit takes the previously generated word  $w_{t-1}$ , hidden state  $H_{t-1}$  and  $H_0$  as input, and generates the next word  $w_t$  that maximizes the probability  $p_t$  over the vocabulary set. That is, the argmax operation is used. However, in the backward calculation, the gradient of  $\partial w_t/\partial p_t$  is calculated as a constant  $1/\tau$  where  $\tau$  is a temperature parameter. Presumably, this approach delivers a biased but smooth gradient signal while maintaining the forward computation exact.

#### **B** Twitter additional results

Figure 6 shows the t-SNE embedding of topic features for speakers in different time zones, demonstrating our self-supervised approach learned reasonable representation to separate different users. Here we use 2000 testing utterances with speakers from 4 different time zones (500 for each). Ta-

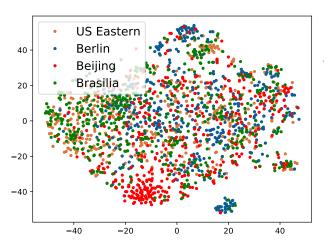


Figure 6: t-SNE embedding of topic features for speakers in different time zones.

ble 7 shows some additional results on controlling the bit, in addition to Table 3.

Context:	A:planning on making a trip to hawaii
	B:lol if only people knew what we are really going there for lolol
	A:hahaha . you we might get judge a little
	B:a little ? probably a lot haha
Plain:	A:it's okay. we should totally go together. we can become friends
	B:lol i'm not sure. i'm going to vegas instead!
	A:haha. i feel you. i just want to get out of all of these.
	B:let's get on our way!!
Electronic	A:it's okay. we should totally go together. i will friend you on fb
on:	
	B:lol sounds good. i will use my phone
	A:sure. let me add you to my contact
	B:i'm going to do that. i will add you too.

Table 7: Examples of bits manipulations. *Plain* denotes without manipulation. X on denotes bit X is activated (set to 1) when generating the response

### C MetaLWOz results

We provide some generated samples for MetaLWOz dataset in Table 8. All compared models use first 4 turns as seed and

generate the remaining 7 turns by taking 4 previous turns as context.

## D Comments on adding Persona Features (-TP)

We indeed observe that the performance gain of CoCon-TP are mostly from adding the topic features (T), presumably because contentful topic information has greater impact on generation, compared to the stylized persona information. However, persona features yield consistent performance improvement over almost all automatic metrics in all our experiments (Table 4,5). We hypothesize that such improvements come from additional utilization of the natural structure of dialogue where two speakers interleave with each other.

## E Comments on the number of the context turns

The topic in a multi-turn dialogue can deviate. However, from a statistical standpoint, the utterances within a dialogue session still demonstrate a relatively high level of topic association. It is difficult to quantify the "contextual appropriateness" vs "topic association." Our motivation is to make the task more challenging, thus encouraging the features to capture more subtle associations rather than direct keywords/phrases that appear in adjacent utterance pair. Since this paper is about maintaining consistency in multi-turn dialog setup, we kept the dialogue with equal or greater than 8 turns (A-B-A-B-A-B-A-B) for Twitter, which still yield a moderate size of dataset with 6M instances (about 5% of the our dataset after removing deleted accounts and tweets and spamming). With access to the FireHose, more could be obtained by incorporating more years of data. For the MetaLWOz dataset, it is 11-turns in nature.

## F Experimental details

The dimension of the LSTM hidden layer is set to 500. We use ADAM as the optimizer with a learning rate of 0.0001. The hyperparameters  $\lambda$  and  $\eta$  are set to 0.01 and 0.1, respectively. For the dimension of feature vectors, we use 100. For the MetaLWOz dataset we use a 50% dropout rate in each CNN layer and  $\lambda$  is set to 0.1. Hyperparameters are selected to maintain discrimination accuracy while reducing  $\mathcal{L}_{ t DeCorr}$ as much as possible. In all our experiment, for the feature extractor  $F(\cdot)$  (blue in Figure 3) and the encoder of the gen-- erator(orange in Figure 3), we use a 3-layer CNN exactly as in (Zhang et al., 2017b). Specifically, a sentence with length T (padded if necessary) is first represented as a 2D matrix of T by E (word embedding size). To encode the sentence into a fixed-dimension vector F, we apply a CNN with 2 strided convolutional layers and 1 fully-connected layer. The filter width, stride, number of filters and word embedding dimensions are set to(filter\_width=5, stride=2, n\_filter=300, embedding\_dim=300). The dimension of final feature vector is 500. The hyperparameters were shared across all architectures in our experiments and chosen based on validation performance of CoCon-T. We experimented on different hyperparameter choices for CoCon-TP, the performance is either comparable or inferior to the hyperparameters that we reported in the paper. The hyperparameters are selected based on a grid search from  $[1, 5e-1, 1e-1, \dots, 5e-5, 1e-5]$ . The grid search over all combinations is computationally intensive. Instead, we first narrowing down to the top-3 candidate values for  $\tau$  (5e-1, 1e-1, 5e-2) and  $\eta$  (1e-2, 5e-3, 1e-3),

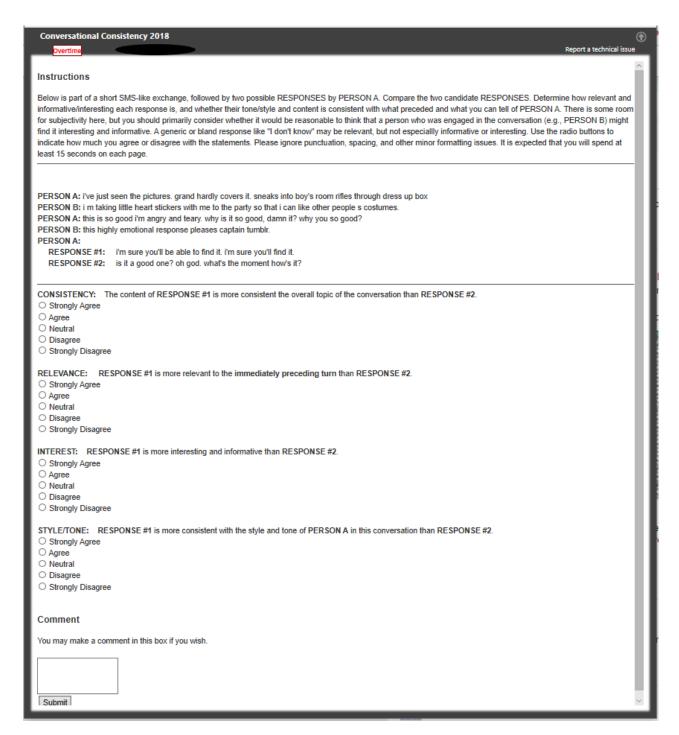


Figure 7: Questionnaire used to elicit pairwise judgments from crowdsourced annotators. Candidate responses were presented in random order.

and performing a 3\*3 search over the combinations. Note that the hyperparameters were shared between CoCon-T and CoCon-TP. In CoCon-TP, there are no persona-specific hyperparameters, thus we simply reuse the best hyperparameters of CoCon-T without additional fine tuning.

#### G Human evaluation

Human evaluation was conducted using the form shown in Figure 7. The two response candidates were presented in random order to the judges, who used a Likert scale to indicate their preferences. To make the questionnaire less abstract to

judges, persona was evaluated in terms of which response better reflected the tone and style of Person A as observable in the prior turns. The distributions of judgments for each of the questions are shown in Tables 9 through 12. We report the Krippendorff's alpha for assessing the inter-rater consistency. Krippendorff's alpha is essentially Fleiss' kappa but can handle variable numbers of annotators, missing entries, different data types. The alpha value is 0.168 with a 95% CI as [0.158, 0.177], based on all annotations.

Source:	A:hello how may i help you ?	Source:	A:hello how may i help you ?
	U:hi i want to look up a play		U:i need to learn more about the rules for tag
	A:what play		A:i can certainly help with game rules . what do you want
	U:what time does the secret miracle opening on ?		to learn about the game tag?
			U:how do i pick the first person to be ' it '
Human:	A:the secret miracle is scheduled for 8 pm on saturday	Human:	A: the first person to ask who is it is considered 'it'.
	U:ok book me for tickets		U:hm, that makes sense. how do you know who is the
	A:ok sure		winner ?
	U:how much is that ?		A:you could keep track of times a person has been ' it '
	A:50		U:oh, okay thanks so much
	U:ok book it now		A:can i help you with something else today?
	A:done		U:no, that will be all today
			A:thanks for taking the time to ask me a question!
CoCon-TP:	A:9:00 PM friday	CoCon-TP:	A:the game of tag is a person who gets tagged the least.
	U:are there any tickets available for saturday there?		U:is there a different rule for tag?
	A:yes tickets do have a deal at 8 pm		A:yes, i can help you with that . there is a lot of tag
	U:ok book me for tickets		U:okay, what is the best time to play the most?
	A:ok ticket		A:the winner is the person that gets tagged the least
	U:reserve for 3 seats please		U:oh, okay, i 'll play that game. thanks for the tips.
	A:okay done		A:you are welcome . i am always here when you need to
			learn.
S2S:	A:it 's on saturday	S2S:	A:the person who asks who is it is declared ' it '.
	U:and what time ?		U:okay , thank you
	A:2:00 PM		A:you 're welcome .
	U:thank you		U:okay, thanks
	A:you 're welcome		A:you 're welcome !
	U:thanks for the help		U:have a good day
	A:you 're welcome		A:you too .

Table 8: Generated Responses for MetaLWOz dataset (in consecutive generation manner)

D	Distribution of Pairwise Topic Consistency Preferences								
Our Method	5	4	3	2	1	Baseline			
CoCon-TP CoCon-TP	12.60% 10.20%	32.60% 29.85%	22.30% 23.10%	24.45% 28.75%	8.05% 8.10%	seq2seq persona			
CoCon-TP	5.10%	16.40%	26.85%	33.70%	17.95%	human			

Table 9: Distribution of topical consistency preferences (%) for our model (CoCon-TP) compared with seq2seq and persona baselines, according to a five-point Likert scale. A 5 indicates a strong preference for CoCon-TP; a 1 indicates strong preference for the alternative system.

Distribution of Pairwise Persona Preferences									
Our Method	5	4	3	2	1	Baseline			
CoCon-TP CoCon-TP	11.30% 8.30%	29.65% 27.35%	29.85% 34.10%	22.50% 23.70%	6.70% 6.55%	seq2seq persona			
CoCon-TP	4.45%	16.90%	33.35%	30.95%	14.35%	human			

Table 10: Distribution of persona consistency preferences (%) for our model (CoCon-TP) compared with seq2seq and persona baselines, according to a five-point Likert scale. A 5 indicates a strong preference for CoCon-TP; a 1 indicates strong preference for the alternative system.

Distribution of Pairwise Relevance Preferences									
Our Method	5	4	3	2	1	Baseline			
CoCon-TP CoCon-TP	13.70% 10.15%		24.15% 25.55%	22.80% 25.85%	8.30% 9.85%	seq2seq persona			
CoCon-TP	4.75%	15.70%	28.10%	31.95%	19.50%	human			

Table 11: Distribution of relevance preferences (%) for our model (CoCon-TP) compared with seq2seq and persona baselines, according to a five-point Likert scale. A 5 indicates a strong preference for CoCon-TP; 1 indicates strong preference for the alternative system.

## **H** Feature disentanglement

We observed that features can heavily correlate (reflected by high Pearson correlation) with each other without the disen-

Distribution of Pairwise Informativeness Preferences								
Our Method	5	4	3	2	1	Baseline		
CoCon-TP CoCon-TP	12.85% 10.45%	29.95% 28.20%		22.60% 23.55%	6.70% 7.60%	seq2seq persona		
CoCon-TP	4.40%	15.05%	29.80%	32.70%	18.05%	human		

Table 12: Distribution of informativeness preferences (%) for our model (CoCon-TP) compared with seq2seq and persona baselines, according to a five-point Likert scale. A 5 indicates a strong preference for CoCon-TP; 1 indicates strong preference for the alternative system.

tangling loss, thus similar "topics" will occupy many feature slots. This is less interpretable (and less efficient in feature vector utilization) as many features are controlling one single "topic". The results without disentangling loss are worse (BLEU-4 drops by 0.2 and NIST drops by 0.13 in CoCon-TP).