Maximizing Stylistic Control and Semantic Accuracy in NLG: Personality Variation and Discourse Contrast

Vrindavan Harrison, Lena Reed, Shereen Oraby, Marilyn Walker
Natural Language and Dialogue Systems Lab
University of California Santa Cruz
Santa Cruz, CA, US
{vharriso, lireed, soraby, mawalker}@ucsc.edu

Abstract

Neural generation methods for task-oriented dialogue typically generate from a meaning representation that is populated using a database of domain information, such as a table of data describing a restaurant. While earlier work focused solely on the semantic fidelity of outputs, recent work has started to explore methods for controlling the style of the generated text while simultaneously achieving semantic accuracy. Here we experiment with two stylistic benchmark tasks, generating language that exhibits variation in personality, and generating discourse contrast. We report a huge performance improvement in both stylistic control and semantic accuracy over the state of the art on both of these benchmarks. We test several different models and show that putting stylistic conditioning in the decoder and eliminating the semantic re-ranker used in earlier models results in more than 15 points higher BLEU for Personality, with a reduction of semantic error to near zero. We also report an improvement from .75 to .81 in controlling contrast and a reduction in semantic error from 16% to 2%.

1 Introduction

Neural encoder-decoder models were originally developed for machine translation (Sutskever et al., 2014; Bahdanau et al., 2014a), but they have also been shown to be successful in related natural language generation (NLG) tasks such as realizing dialogue system utterances from meaning representations (MRs) as shown for the restaurant domain in Table 1 (Dusek and Jurcícek, 2016). Recent work in neural NLG has shown that stylistic control is an important problem in its own right: it is needed to address a well-known limitation of such models, namely that they reduce the stylistic variation seen in the input, and thus produce outputs that tend to be dull and repetitive (Li et al., 2016). Here we compare different methods for directly controlling stylistic variation when generating from MRs, while simultaneously achieving high semantic accuracy.

Meaning Representation
name[Browns Cambridge], eatType[coffee shop], food[Italian], customerRating[average], area[riverside], familyFriendly[yes], near[Crowne Plaza Hotel]
E2E Dataset
Browns Cambridge is near Crowne Plaza Hotel. Browns Cambridge has a three star rating. Browns Cambridge is a family coffee shop.
Personality: Conscientious
Did you say Browns Cambridge? Well, i see, I think that it is a coffee shop, also it has a decent rating, and it is near Crowne Plaza Hotel kid friendly in riverside and an Italian place.
Personality: Disagreeable
Come on, I am not sure. I mean Browns Cambridge is an Italian place, also it has a damn average rating. It is near Crowne Plaza Hotel.
Personality: Unconscientious
Oh God I don't know! Browns Cambridge is a coffee shop, also it is family friendly near Crowne Plaza Hotel, also it is an Italian place in riverside, also it has like, a decent rating.

Table 1: Sample meaning representation with a realization from the E2E Challenge Dataset and three stylistic personality realizations.

Tables 1 and 2 illustrate the two stylistic benchmark datasets that form the basis of our experimental setup. Table 1 shows an example MR with three surface realizations: the E2E realization does not target a particular personality, while the other two examples vary stylistically according to linguistic profiles of personality type (Pennebaker and King, 1999; Furnham, 1990; Mairesse and Walker, 2011). Table 2 shows an example MR with two surface realizations that vary stylistically according to whether the discourse contrast relation is used (Nakatsu and White, 2006; Howcroft et al., 2013). Both of these benchmarks provide parallel data that supports experiments that hold constant the underlying meaning of an utterance, while varying the style of the output text. In

Proceedings of the 1st Workshop on Discourse Structure in Neural NLG, pages 1–12
Tokyo, Japan, November 1, 2019.

Meaning Representation
name[Brown's Cambridge], food[Italian], customerRating[3 out of 5], familyFriendly[no], price[moderate]
With Contrast Relation
Browns Cambridge is an Italian restaurant with average customer reviews and ***reasonable prices, but it is not child-friendly.***
Without Contrast Relation
Browns Cambridge serves Italian food in moderate price range. It is not kid friendly and the customer rating is 3 out of 5.

Table 2: A sample meaning representation with contrastive and non-contrastive surface realizations.

contrast, other tasks that have been used to explore methods for stylistic control such as machine translation or summarization (known as text-to-text generation tasks) do not allow for such a clean separation of meaning from style because the inputs are themselves surface forms.

We describe three methods of incorporating stylistic information as *side constraints* into an RNN encoder-decoder model, and test each method on both the personality and contrast stylistic benchmarks. We perform a detailed comparative analysis of the strengths and weaknesses of each method. We measure both semantic fidelity and stylistic accuracy and quantify the tradeoffs between them. We show that putting stylistic conditioning in the decoder, instead of in the encoder as in previous work, and eliminating the semantic re-ranker used in earlier models results in more than 15 points higher BLEU for Personality, with a reduction of semantic error to near zero. We also report an improvement from .75 to .81 in controlling contrast and a reduction in semantic error from 16% to 2%. To the best of our knowledge, no prior work has conducted a systematic comparison of these methods using such robust criteria specifically geared towards controllable stylistic variation. We delay a detailed review of prior work to Section 4 when we can compare it to our own.

2 Models and Variants

In the recent E2E NLG Challenge shared task, models were tasked with generating surface forms from structured meaning representations(Duek et al., 2019). The top performing models were all RNN encoder-decoder systems. Our model also follows a standard RNN Encoder–Decoder model (Sutskever et al., 2014; Bahdanau et al., 2014a) that maps a source sequence (the input MR) to a target sequence.

2.1 Model

Our model represents an MR as a sequence $x = (x_1, x_2, \ldots x_n)$ of slot-value pairs. The generator is tasked with generating a surface realization which is represented as a sequence y of tokens $y_1, y_2, \ldots y_m$. The generation system models the conditional probability $p(y|x)$ of generating the surface realization y from some meaning representation x. Thus, by predicting one word at a time, the conditional probability can be decomposed into the conditional probability of the next token in the output sequence:

$$p(y|x) = \prod_{t=1}^{m} p(y_t|y_1, y_2, \ldots y_{t-1}, x) \ . \quad (1)$$

We are interested in exercising greater control over the characteristics of the output sequence by incorporating *side constraints* into the model (Sennrich et al., 2016). The side constraints $\mathbf{c}$ act as an additional condition when predicting each token in the sequence. In this case, the conditional probability of the next token in the output sequence is given by:

$$p(y|x, \mathbf{c}) = \prod_{t=1}^{m} p(y_t|y_1, y_2, \ldots y_{t-1}, x, \mathbf{c}) \ . \quad (2)$$

In Section 2.2 we describe three methods of computing $p(y|x, \mathbf{c})$.

Encoder. The model reads in an MR as a sequence of slot-value pairs. Separate vocabularies for slot-types and slot values are calculated in a pre-processing step. Each slot type and slot value are encoded as one-hot vectors which are accessed through a table look-up operation at run-time. Each slot-value pair is encoded by first concatenating the slot type encoding with the encoding of its specified value. Then the slot-value pair is encoded with an RNN encoder. We use a multi-layer bidirectional LSTM (Hochreiter and Schmidhuber, 1997) to encode the input sequence of MR slot-value pairs. The hidden state $\bar{h_i}$ is represented as the concatenation of the forward state $\overrightarrow{h_i}$ and backward state $\overleftarrow{h_i}$. Specifically, $\bar{h_i} = (\overrightarrow{h_i}, \overleftarrow{h_i})$.

Decoder. The decoder is a uni-directional LSTM. Attention is implemented as in (Luong et al., 2015). We use a global attention where the attention scores between two vectors a and b are calculated as $a^T \mathbf{W} b$, where $\mathbf{W}$ is a model parameter learned during training.

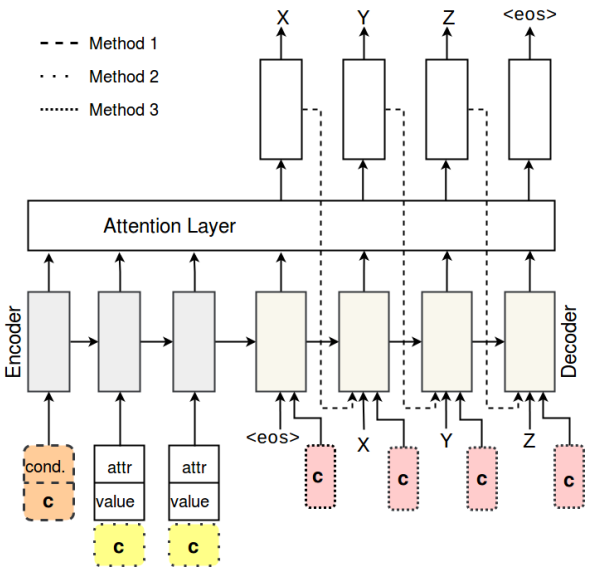

Figure 1: Attentional Encoder-Decoder architecture with each of the three side constraint implementations shown. The output sequence X, Y, Z is being generated from an MR represented as an input sequence of attribute value pairs.

2.2 Side Constraints

Recent work has begun to explore methods for stylistic control in neural language generation, but there has been no systematic attempt to contrast different methods on the same benchmark tasks and thereby gain a deeper understanding of which methods work best and why. Here, we compare and contrast three alternative methods for implementing side constraints in a standard encoder-decoder architecture. The first method involves adding slot-value pairs to the input MR, and the second involves extending the slot-value encoding through a concatenation operation. In the third method, side constraints are incorporated into the model by modifying the decoder inputs. The three side constraint implementation methods are shown simultaneously in Figure 1. The orange area refers to Method 1, the yellow areas corresponds to Method 2, and the red areas corresponds to Method 3.

Method 1: Token Supervision. This method provides the simplest way of encoding stylistic information by inserting an additional token that encodes the side constraint into the sequence of tokens that constitute the MR (Sennrich et al., 2016). We add a new slot type representing `side-constraint` to the vocabulary of slot-types, and new entries for each of the possible side

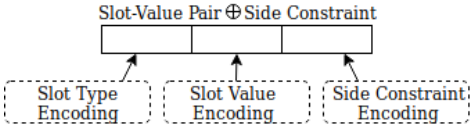

Figure 2: Slot-value encoding extended with constraint.

constraint values to the vocabulary of slot values.

Method 2: Token Features. This method incorporates side constraints through use of a slot-value pair feature. First we construct a vector representation c that contains the side constraint information. Normally the individual slot-value pair encodings are built by concatenating the slot-type with the slot-value as with Method 1. We modify each slot-value pair encoding of the MR by extending it with c, as seen in Figure 2.

Method 3: Decoder Conditioning. This method incorporates side constraint information into the generation process by adding additional inputs to the LSTM decoder. Traditionally, at the t-th time step a LSTM decoder takes two inputs. One input is the previous ground truth token's embedding w_{t-1}, and the other is a context vector d_t which is an attention-weighted average of the encoder hidden states. A vector c containing side constraint information is provided to the decoder as a third input. Thus at each time step the decoder's hidden state $\tilde{h}_i$ is calculated as

$$\tilde{h}_i = \text{LSTM}([w_{t-1}; d_t; c])\,. \tag{3}$$

3 Experiments: Varying Personality and Discourse Structure

We perform two sets of experiments using two stylistic benchmark datasets: one for personality, and one for discourse structure, i.e., contrast. In both cases, our aim is to generate stylized text from meaning representations (MRs). In the personality experiments, the generator's goal is to vary the personality style of the output and accurately realize the MR. The personality type is the side constraint that conditions model outputs, and is represented using a 1-hot encoding for the models that use side constraint Methods 2 and 3. For the sake of comparison, we also train a model that does not use conditioning (NoCon). In the discourse contrast experiments, the generator's goal is to control whether the output utterance uses the discourse contrast relation. The side constraint is

Personality	Realization
Meaning Representation	name[The Eagle], eatType[coffee shop], food[English], priceRange[cheap], customer rating[average], area[riverside], familyFriendly[yes], near[Burger King]
Agreeable	You want to know more about The Eagle? Yeah, ok it has an average rating, it is a coffee shop and it is an English restaurant in riverside, quite cheap near Burger King and family friendly.
Disagreeable	Oh god I mean, I thought everybody knew that The Eagle is cheap with an average rating, it's near Burger King, it is an English place, it is a coffee shop and The Eagle is in riverside, also it is family friendly.
Conscientious	I think that The Eagle is a coffee shop, it has an average rating and it is somewhat cheap in riverside and an English restaurant near Burger King. It is rather kid friendly.
Unconscientious	Yeah, I don't know. Mmhm ... The Eagle is a coffee shop, The Eagle is cheap, it's kind of in riverside, it is an English place and The Eagle has an average rating. It is kind of near Burger King.
Extravert	The Eagle is a coffee shop, you know, it is an English place, family friendly in riverside and cheap near Burger King and The Eagle has an average rating friend!

Table 3: Model outputs for each personality style for a fixed Meaning Representation (MR). The model was trained using control Method 3.

a simple boolean: contrast, or no contrast. The model is tasked with learning 1) which category of items can potentially be contrasted (e.g., *price* and *rating* can appear in a contrast relation but *name* can not), and 2) which values are appropriate to contrast (i.e., items with polar opposite values).

All models are implemented using PyTorch and OpenNMT-py[1](Klein et al., 2017). We use Dropout (Srivastava et al., 2014) of 0.1 between RNN layers. Model parameters are initialized using Glorot initialization (Glorot and Bengio, 2010) and are optimized using stochastic gradient descent with mini-batches of size 128. Beam search with three beams is used during inference. We implement multiple models for each experiment using the methods for stylistic control discussed in Section 2.2. We tune model hyper-parameters on a development dataset and select the model of lowest perplexity to evaluate on a test dataset. All models are trained using lower-cased and delexicalized reference texts. The sample model outputs we present have been re-capitalized and re-lexicalized using a simple rule based script. Further details on model implementation, hyper-parameter tuning, and data processing are provided as supplementary material.

3.1 Benchmark Datasets and Experiments

Personality Benchmark. This dataset provides multiple reference outputs for each MR, where the style of the output varies by personality type (Oraby et al., 2018b).[2] The styles belong to the Big Five personality traits: agreeable, disagreeable, conscientious, un-conscientious, and extrovert, each with a stylistically distinct linguistic profile (Mairesse and Walker, 2010a; Furnham, 1990). Example model outputs for each personality on a fixed MR are in Table 3.

The dataset consists of 88,855 train examples and 1,390 test examples that are evenly distributed across the five personality types. Each example consists of a (MR, personality-label, reference-text) tuple. The dataset was created using the MRs from the E2E Dataset (Novikova et al., 2017) and reference texts synthesized by PERSONAGE (Mairesse and Walker, 2010b), a statistical language generator capable of generating utterances that vary in style according to psycho-linguistic models of personality. The statistical generator is configured using 36 binary parameters that target particular linguistic constructions associated with different personality types. These are split into *aggregation operations* that combine individual propositions into larger sentences, and *pragmatic markers* which typically modify some expression within a sentence, e.g. *tag questions* or *in-group markers*. A subset of these are illustrated in Table 4: see Oraby et al. (2018b) for more detail.

We conduct experiments using two control configurations that differ in the granularity of control that they provide. We call the first configuration *course-grained* control, and the model is conditioned using a single constraint: the personality label. The second configuration, called *fine-grained* control, conditions the model using the personality label and Personage's 36 binary control parameters as illustrated by Table 4, which provide fine-grained information on the desired style of the out-

[1] github.com/OpenNMT/OpenNMT-py

[2] nlds.soe.ucsc.edu/stylistic-variation-nlg

Attribute	Example
AGGREGATION OPERATIONS	
"WITH" CUE	*X is in Y, with Z.*
CONJUNCTION	*X is Y and it is Z. & X is Y, it is Z.*
"ALSO" CUE	*X has Y, also it has Z.*
PRAGMATIC MARKERS	
ACK_JUSTIFICATION	*I see, well*
ACK_YEAH	*yeah*
CONFIRMATION	*let's see ....., did you say X?*
DOWN_KIND_OF	*kind of*
DOWN_LIKE	*like*
EXCLAIM	*!*
GENERAL SOFTENER	*sort of, somewhat, quite, rather*
EMPHASIZER	*really, basically, actually, just*
TAG QUESTION	*alright?, you see? ok?*

Table 4: Example Aggregation and Pragmatic Operations

put text. The stylistic control parameters are not updated during training. When operating under fine-grained control, for side constraint Methods 2 and 3, the 1-hot vector that encodes personality are extended with dimensions for each of the 36 control parameters. For Method 1 we insert 36 tokens, one for each parameter, to the beginning of each input sequence, in addition to the single token that represents personality label.

Contrast Benchmark. This dataset provides reference outputs for 1000 MRs, where the style of the output varies by whether or not it uses the discourse contrast relation.[3] Contrast training set examples are shown in Table 2.

The contrast dataset is based on 15,000 examples from the E2E generation challenge, which consists of 2,919 contrastive examples and 12,079 examples without contrast.[4] We split the dataset into train and development subsets using a 90/10 split ratio. The test data is composed of a set of 500 MRs that contain attributes that can be contrasted, whose reference outputs use discourse-contrast (Reed et al., 2018). The test set also contains a set of 500 MRs that were selected from the E2E development set that do not use discourse-contrast. We crowd-sourced human-generated references for the contrastive test set, and used the references from the E2E dataset for the non-contrastive test set.[5]

[3] nlds.soe.ucsc.edu/sentence-planning-NLG

[4] www.macs.hw.ac.uk/InteractionLab/E2E/

[5] We will make our test and partitions of training data available to the research community if this paper is accepted.

3.2 Results

For both types of stylistic variation, we evaluate model outputs using automatic metrics targeting semantic quality, diversity of the outputs, and the type of stylistic variation the model is attempting to achieve. We also conduct two human evaluations. In the tables and discussion that follow, we refer to the models that employ each of the side constraint methods, e.g., Methods 1, 2, and 3, described in Section 2.2, using the monikers M{1,2,3}. The model denoted NoCon refers to a model that uses no side constraint information. Sample model outputs from the personality experiments are shown in Table 3. The outputs are from the M3 model when operating under the fine grained control setting. Outputs from model M2 of the contrast experiment are shown in Table 8.

3.2.1 Semantic Quality

Model	BLEU	SER	H	AGG	PRAG
Oraby et al. (2018b)					
NoCon	27.74	-	7.87	.56	.08
coarse	34.64	-	8.47	.64	.48
fine	37.66	-	8.58	.71	.55
This Work					
Train	-	-	9.34	-	-
NoCon	38.45	**0**	7.70	.44	.14
coarse control					
M1	49.04	0.000	8.49	.57	.51
M2	48.10	0.002	8.52	.62	.50
M3	49.06	0.009	8.50	.60	.50
fine control					
M1	55.30	0.004	8.77	.82	.94
M2	52.29	0.103	**8.80**	.84	.93
M3	**55.98**	0.014	8.74	.84	.93

Table 5: Automatic evaluation on Personality test set. *course* and *fine* refer to the specificity of the control configuration.

First, we measure general similarity between model outputs and gold standard reference texts using BLEU, calculated with the same evaluation script[6] as Oraby et al. (2018b). For the personality experiment, the scores for each conditioning method and each control granularity are shown in Table 5, along with the results reported by Oraby et al. (2018b). For the contrast experiment, the scores for each conditioning method are shown in Table 6, where we refer to the model and results of Reed et al. (2018) as *M-Reed*. Reed et al. (2018) do not report BLEU or Entropy (H) measures.

We first discuss the baselines from previous work on the same benchmarks. Interestingly, for

[6] github.com/tuetschek/e2e-metrics

Personality, our NoCon model gets a huge performance improvement of more than 11 points in BLEU (27.74 → 38.45) over results reported by Oraby et al. (2018a). We note that while the underlying architecture behind our experiments is similar to the baseline described by Oraby et al. (2018a), we experiment with different parameters and attention mechanisms. Reed et al. (2018) and Oraby et al. (2018b) also use an LSTM encoder-decoder model with attention, but they both implement their models using the TGen[7](Dušek and Jurcicek, 2016) framework with its default model architecture. TGen uses an early version of TensorFlow with different initialization methods, and dropout implementation. Moreover, we use a different one-hot encoding of slots and their values, and we implement attention as in Luong et al. (2015), whereas TGen uses Bahdanau et al. (2014b) attention by default. Side constraints are incorporated into the TGen models in two ways: 1) using a new dialogue act type to indicate the side constraints, and 2) a feed-forward layer processes the constraints and, during decoding, attention is computed over the encoder hidden states and the hidden state produced by the feed-forward layer. The TGen system uses beam-search and an additional output re-ranking module.

We now compare the performance of our own model results in Table 5. As would be expected, NoCon has the lowest performance overall of all models, with a BLEU of 38.45. With both coarse control and fine-grained control, M3 and M2 are the highest and lowest performers, respectively. For the contrast experiment, M2 and M3 have very similar values for all rows of Table 6. M2 has the highest BLEU score of 17.32 and M3 has 17.09. M1 is consistently outperformed by both M2 and M3. All side constraint models outperform NoCon. We note that the contrast task achieves much lower scores on BLEU. This maybe due to the relatively small number of contrast examples in the training set, but it is also possible that this indicates the large variety of ways that contrast can be expressed, rather than poor model performance. We show in a human evaluation in Section 3.2.2 that the contrast examples are fluent and stylistically interesting.

A comparison of our results versus those reported by Oraby et al. (2018b) are also shown in Table 5. Note that our model has an over 14 point margin of improvement in BLEU score when using coarse control and a more than 18 point improvement when using fine-grained control. Our models can clearly use the conditioning information more effectively than earlier work.

Model	BLEU	SER	H
Train	-		10.68
Contrast Data			
M-Reed	-	.16	-
NoCon	15.80	**.053**	**8.09**
M1	16.58	.055	8.08
M2	**17.32**	.058	8.03
M3	17.09	.058	7.93
Non Contrast Data			
NoCon	26.58	.025	7.67
M1	26.58	.023	7.56
M2	26.35	.017	7.68
M3	26.04	.035	7.40

Table 6: Automatic evaluation on Contrast test set.

Slot Error Rate. While the n-gram overlap metrics are able to measure general similarity between gold references and model outputs, they often do not do a good job at measuring semantic accuracy. Slot error rate (SER)(Wen et al., 2015; Reed et al., 2018) is a metric similar to word error rate that measures how close a given realization adheres to its MR. SER[8] is calculated using the slot aligner released[9] by Juraska et al. (2018) to count the number of attributes (slots) and their values that correctly (and incorrectly) occur in a given surface realization. Please refer to Supplementary Materials, Section A.1 for the definition of SER.

We evaluate each model using SER with results in Tables 5 and 6. We first note that all the SERs for both tasks are extremely low and that only M2 under fine control performs worse with an SER of .10. The models are clearly learning to realize the intended MRs. M1 has the best SER scores in all experiment conditions. In the contrast experiment, M2 and M3 are practically equivalent.

Model	Acc	Contrast Attempts
M-Reed	.75	422
M1	.74	437
M2	.79	485
M3	.81	474

Table 7: Contrast accuracy out of 500 examples.

[7] github.com/UFAL-DSG/tgen

[8] A formal definition of SER is provided in the supplementary materials.

[9] github.com/jjuraska/slug2slug

Meaning Representation	Realization
name[Aromi], eatType[restaurant], rating[low], familyFriendly[yes]	*Aromi is a **family friendly restaurant but the customer rating is low**.*
name[Fitzbillies], cuisine[English], price[more than $30], eatType[pub], familyFriendly[yes]	*Fitzbillies is a pub that serves English food, **is children friendly, but the price range is more than $30**.*
name[Clowns], price[more than $30], rating[high], familyFriendly[no], near[Clare Hall]	*Clowns is near Clare Hall. It has a **high customer rating but is not child friendly**.*
name[Cotto], cuisine[English], location[riverside], price[high], eatType[coffee shop], rating[5 out of 5], near[The Portland Arms]	*Cotto is a English coffee shop near The Portland Arms in the riverside. It has **a high price range but a customer rating of 5 out of 5**.*

Table 8: Sample outputs from model M2 with contrast relation in bold.

3.2.2 Quality in Variation

In the previous section we tested the ability of the side constraint models to produce semantically accurate outputs. In this section we evaluate the extent to which the side constraint models produce stylistically varied texts. We evaluate variation using two measures: 1) Entropy, and 2) counts on model outputs for particular stylistic targets.

Entropy. Our goal is NLG models that produce stylistically rich, diverse outputs, but we expect that variation in the training data will be averaged out during model training. We quantify the amount of variation in the training set, and also in the output references from the test set MRs using Entropy[10], H, where a larger entropy value indicates a larger amount of linguistic variation preserved in the test outputs.

The results are shown in the H column of Tables 5 and 6. For the personality experiment, the training corpus has 9.34 entropy and none of the models are able to match its variability. When using fine-grained control M2 does the best with 8.52 but all side constraint models are within 0.03. When using coarse control M2 has the highest entropy with 8.80. Our models with fine control outperform Oraby et al. (2018b) in terms of entropy. For the contrast experiment, NoCon has the highest entropy at 8.09, but the differences are small.

Counts of Stylistic Constructions. Entropy measures variation in the corpus as a whole, but we can also examine the model's ability to vary its outputs in agreement with the stylistic control parameters. Contrast accuracy measures the ratio of valid contrast realizations to the number of contrasts attempted by the model. We determine valid contrasts using the presence of polar opposite values in the MR and then inspecting realization of those values in the model output.

[10]A formal definition of our Entropy calculation is provided with the supplementary materials.

Table 7 shows the results. The row labeled M-Reed refers to the results reported by Reed et al. (2018). NoCon rarely attempts contrast because there is no way to motivate it to do so, and it therefore produces no contrast. Contrast attempts are out of 500 and M2 has the most at 485. In terms of contrast accuracy M3 is the best with 81%.

When comparing our model performance to M-Reed, models M{1,2,3} make more contrast attempts. M1 and M-Reed have similar contrast accuracy with 74% and 75%, respectively. The higher performance of our models is particularly impressive since the M-Reed models see roughly 7k contrast examples during training, which is twice the amount that our models see.

For personality, we examine each model's ability to vary its outputs in agreement with the stylistic control parameters by measuring correlations between model outputs and test reference texts in the use of the aggregation operations and pragmatic markers, two types of linguistic constructions illustrated in Table 4, and associated with each personality type. The results for these linguistic constructions over all personality types are shown in the last two columns (Agg, Prag) of Table 5. The supplementary material provides details for each personality. Our results demonstrate a very large increase in the correlation of these markers between model outputs and reference texts compared to previous work, and also further demonstates the benefits of fine-grained control, where we achieve correlations to the reference texts as high as .94 for pragmatic markers and as high as .84 for aggregation operations.

Methods Comparison. The results in Tables 5 and 7 reveal a general trend where model performance in terms of BLEU and entropy increases as more information is given to the model as side constraints. At the same time, the slot error rates are somewhat higher, indicating the difficulty of

simultaneously achieving both high semantic and stylistic fidelity. Our conclusion is that Method 3 performs the best at controlling text style, but only when it has access to a large training dataset, and Method 2 performs better in situations where training data is limited.

Human evaluation. We perform human evaluation of the quality of outputs for the M3 model with a random sample of 50 surface realizations for each personality, and 50 each for contrast and non-contrast outputs for a total of 350 examples. Three annotators on Mechanical Turk rate each output for both interestingness and fluency (accounting for both grammaticality and naturalness) using a 1-5 Likert scale.

Human evaluation results are shown in Table 9 for the personality experiment and Table 10 for contrast. The tables show average annotator rating in each category. For the personality outputs, each personality has similar fluency ratings with Conscientious slightly higher. The model outputs for the contrast relation have higher average ratings for Fluency than the non-contrastive realizations. For interestingness, we compare both the personality styles and the contrastive style to the basic style without contrast. The results show that non-contrast (3.07), the vanilla style, is judged as significantly less interesting than the personality styles (ranging from 3.39 to 3.51) or the use of discourse contrast (3.45) (p-values all less than .01).

	Con.	Dis.	Agr.	Ext.	Unc.	avg
Fluent	3.77	3.38	3.53	3.38	3.35	3.48
Interest	3.39	3.40	3.51	3.46	3.45	3.44

Table 9: Human evaluation results for personality.

	Non-contrast	Contrast
Fluent	4.21	4.38
Interest	3.07	3.45

Table 10: Human evaluation results for discourse contrast.

4 Related Work

Stylistic control is important as a way to address a well-known limitation of vanilla neural NLG models, namely that they reduce the stylistic variation seen in the input, and thus produce outputs that tend to be dull and repetitive (Li et al., 2016). The majority of other work on stylistic control has been done in a text-to-text setting where MRs and corpora with fixed meaning and varying style are not available (Fan et al., 2017; Iyyer et al., 2018; Wiseman et al., 2018; Ficler and Goldberg, 2017). Sometimes variation is evaluated in terms of model performance in some other task, such as machine translation or summarization. Herzig et al. (2017) also control personality in the context of text-2-text generation in customer care dialogues. Kikuchi et al. (2016) control output sequence length by adding a remaining-length encoding as extra input to the decoder. Sennrich et al. (2016) control linguistic honorifics in the target language by adding a special social formality token to the end of the source text. Hu et al. (2017) control sentiment and tense (past, present, future) in text2text generation of movie reviews. Ficler and Goldberg (2017) describe a conditioned language model that controls variation in the stylistic properties of generated movie reviews.

Our work builds directly on the approach and benchmark datasets of Reed et al. (2018) and Oraby et al. (2018b). Here we compare directly to the results of Oraby et al. (2018b), who were the first to show show that a sequence-to-sequence model can generate utterances from MRs that manifest a personality type. Reed et al. (2018) also develop a neural model for a controllable sentence planning task and run an experiment similar to our contrast experiment. Here, we experiment extensively with different control methods and present large performance improvements on both tasks.

5 Conclusion

We present three different models for stylistic control of an attentional encoder-decoder model that generates restaurant descriptions from structured semantic representations using two stylistic benchmark datasets: one for personality variation and the other for variation in discourse contrast. We show that the best models can simultaneously control the variation in style while maintaining semantic fidelity to a meaning representation. Our experiments suggest that overall, incorporating style information into the decoder performs best and we report a large performance improvement on both benchmark tasks, over a large range of metrics specifically designed to measure semantic fidelity along with stylistic variation. A human evaluation shows that the outputs of the best models are judged as fluent and coherent and that the stylistically controlled outputs are rated significantly more interesting than more vanilla outputs.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014a. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473 [cs, stat]*. ArXiv: 1409.0473.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014b. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Jean-Marc Dewaele and Adrian Furnham. 1999. Extraversion: The unloved variable in applied linguistic research. *Language Learning*, 49(3):509–544.

Ondřej Dušek and Filip Jurcicek. 2016. A context-aware natural language generator for dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 185–190.

Ondrej Dusek and Filip Jurcícek. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *CoRR*, abs/1606.05491.

Ondej Duek, Jekaterina Novikova, and Verena Rieser. 2019. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *arXiv:1901.07931 [cs]*. ArXiv: 1901.07931.

Angela Fan, David Grangier, and Michael Auli. 2017. Controllable abstractive summarization. *arXiv:1711.05217 [cs]*. ArXiv: 1711.05217.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, page 94104. Association for Computational Linguistics.

Adrian Furnham. 1990. Language and personality. In H. Giles and W. Robinson, editors, *Handbook of Language and Social Psychology*. Winley.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, page 249256.

Jonathan Herzig, Michal Shmueli-Scheuer, Tommy Sandbank, and David Konopnicki. 2017. Neural response generation for customer service based on personality traits. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 252–256.

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

David M Howcroft, Crystal Nakatsu, and Michael White. 2013. Enhancing the expression of contrast in the sparky restaurant corpus. *ENLG 2013*, page 30.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596, International Convention Centre, Sydney, Australia. PMLR.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1:18751885.

Juraj Juraska, Panagiotis Karagiannis, Kevin Bowden, and Marilyn Walker. 2018. A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 152–162.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, page 13281338. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Penelope Levinson, Penelope Brown, Stephen C Levinson, and Stephen C Levinson. 1987. *Politeness: Some universals in language usage*, volume 4. Cambridge university press.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

F. Mairesse and M.A. Walker. 2010a. Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction*, pages 1–52.

François Mairesse and Marilyn A Walker. 2010b. Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction*, 20(3):227–278.

Francois Mairesse and Marilyn A. Walker. 2011. Controlling user perceptions of linguistic style: Trainable generation of personality traits. *Computational Linguistics*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60.

Crystal Nakatsu and Michael White. 2006. Learning to say it well: Reranking realizations by predicted synthesis quality. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1113–1120.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrcken, Germany. Association for Computational Linguistics.

Jon Oberlander and Alastair J Gill. 2004. Individual differences and implicit language: personality, parts-of-speech and pervasiveness. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 26.

Shereen Oraby, Lena Reed, TS Sharath, Shubhangi Tandon, and Marilyn Walker. 2018a. Neural multivoice models for expressing novel personalities in dialog. *Proc. Interspeech 2018*, pages 3057–3061.

Shereen Oraby, Lena Reed, Shubhangi Tandon, TS Sharath, Stephanie Lukin, and Marilyn Walker. 2018b. Controlling personality-based stylistic variation with neural natural language generators. In *SIGDIAL*.

J. W. Pennebaker and L. A. King. 1999. Linguistic styles: Language use as an individual difference. *Journal of Personality and Social Psychology*, 77:1296–1312.

Lena Reed, Shereen Oraby, and Marilyn Walker. 2018. Can neural generators for dialogue learn sentence planning and discourse structuring? *arXiv:1809.03015 [cs]*. ArXiv: 1809.03015.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, page 3540. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):19291958.

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

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2018. Learning neural templates for text generation. *arXiv:1808.10122 [cs]*. ArXiv: 1808.10122.

A Supplementary Materials: Maximizing Stylistic Control and Semantic Accuracy in Dialogue Generation: Conditional Decoding for Personality Variation and Discourse Contrast

A.1 Calculating Slot Error Rate

Multiple methods of measuring SER have been proposed (Wen et al., 2015; Reed et al., 2018). In this work we use a method similar to the one described by Reed et al. (2018). First, we define the following types of errors: substitutions (realizing an attribute with an incorrect value), deletions (failing to mention an attribute), repeats, and hallucinations (mentioning an attribute that does not appear in the MR).

The SER score for a given (MR, text realization) pair is calculated by first calculating S, D, R, and $\tilde{H}$, which are the amounts of substitutions, deletions, repeats, and hallucinations, respectively. The SER formula is then given as:

$$\text{SER} = \frac{S + D + R + \tilde{H}}{N} \tag{4}$$

where N is the number of slots in the MR. Note that using this method can result in SER values greater than one, since it is possible for there to be more errors than slots in the MR.

A.2 Calculating Entropy

To calculate Shannon Text Entropy H, we first construct the corpus vocabulary V of all unigrams, bigrams, and trigrams. Then H is given by the equation

$$H = -\sum_{a \in V} \frac{k_a}{N} \cdot \log_2(\frac{k_a}{N}) \tag{5}$$

where N is the sum total of occurrences for all terms in V, and k_a is the number of occurrences for the term a.

A.3 Model Implementation Details

Model Implementation. All models are implemented using PyTorchand OpenNMT-py[11] (Klein et al., 2017). We use Dropout (Srivastava et al., 2014) of 0.1 between RNN layers. Model parameters are initialized using Glorot initialization (Glorot and Bengio, 2010) and are optimized using stochastic gradient descent with mini-batches of size 128. Beam search with three beams is used during inference. We implement multiple models for each experiment using the methods for stylistic control discussed in Section 2.2. We tune model hyper-parameters on a development dataset and select the model of lowest perplexity to evaluate on a test dataset. All models are trained using lower-cased and de-lexicalized reference texts. The sample model outputs we present have been re-capitalized and re-lexicalized using a simple rule based script.

Hyper Parameter Tuning. Hyper parameters are tuned using a grid search over the following parameter space:

- RNN layers over the range [1, 2]
- RNN size over the range [150, 200, 250, 300]

We tune the number RNN layers and RNN size by training a model for each combination of layers and RNN size (8 models). We use the model of lowest development dataset perplexity to evaluate on the test dataset.

This parameter tuning process is performed for each of the side constraint methods and style parameter configuration (fine control, coarse control). The resulting hyper parameter values are shown in Table 11

Model	RNN layers	RNN size
NoCon	2	150
coarse control		
M1	1	200
M2	1	200
M3	2	150
fine control		
M1	1	200
M2	2	200
M3	1	200

Table 11: Model hyper-parameter values.

A.4 Data Processing

The data is pre-processed using Stanford CoreNLP (Manning et al., 2014).

A.5 Linguistic constructions: Pragmatic Markers and Aggregation Operations

Psycholinguistic studies have shown these markers to be indicative of the language of people with different personality traits (Pennebaker and King,

[11] github.com/OpenNMT/OpenNMT-py

1999; Furnham, 1990). For example, the use of pragmatic markers has been shown to effect perceptions of personality traits such as politeness, friendliness, extraversion, and enthusiasm (Oberlander and Gill, 2004; Levinson et al., 1987; Dewaele and Furnham, 1999). Using a method similar to Oraby et al. (2018b), we count the occurrences of pragmatic markers and aggregation operations in the model outputs. Then we average the counts within each personality category and calculate the Pearson correlation between the model output averages and the gold reference text averages.

The Pearson correlation r for pragmatic markers can be seen in Table 12. All values of r are significant with p-values less than 0.01. The model with no side constraints has $r \leq 0.17$ for all personalities except for conscientious with $r = 0.81$. This suggests that the un-constrained model picks one personality to optimize – conscientious in this case. For both control granularities each of the side constraint models have similar performance. Table 12 also shows the correlation results reported by Oraby et al. (2018b) where we observe a marked improvement in the pragmatic marker correlations of our models compared to theirs.

Pearson correlations for aggregation operations are shown in Table 13. Again, the test for correlation results in p-values less than 0.01 for each personality type. Here, the Token model of Oraby et al. (2018b) outperforms all three of our models when conditioning on only the personality label (coarse control).

Model	AGR	CON	DIS	EXT	UNC	avg
Oraby et al						
NoSup	0.05	0.59	-0.07	-0.06	-0.11	.08
Token	0.35	0.66	0.31	0.57	0.53	.48
Context	0.28	0.67	0.40	0.76	0.63	.55
This Work - coarse control						
NoCon	.17	.81	-.08	-.08	-.11	.14
M1	.44	.81	.17	.79	.32	.51
M2	.44	.81	.17	.83	.27	.50
M3	.40	.81	.14	.83	.31	.50
This Work - fine control						
M1	.87	.94	.98	.99	.90	.94
M2	.87	.94	.98	.99	.88	.93
M3	.87	.93	.97	.99	.90	.93

Table 12: Correlations between test examples and model outputs for pragmatic markers.

Model	AGR	CON	DIS	EXT	UNC	avg
Oraby et al						
NoSup	0.78	0.80	0.13	0.42	0.69	.56
Token	0.74	0.74	0.57	0.56	0.60	.64
Context	0.83	0.83	0.55	0.66	0.70	.71
This Work - coarse control						
NoCon	0.70	0.73	-0.19	0.35	0.60	.44
M1	0.67	0.70	0.58	0.56	0.36	.57
M2	0.61	0.70	0.58	0.60	0.60	.62
M3	0.64	0.68	0.58	0.59	0.49	.60
This Work - fine control						
M1	0.84	0.91	0.78	0.81	0.78	.82
M2	0.89	0.92	0.78	0.79	0.84	.84
M3	0.86	0.91	0.79	0.82	0.81	.84

Table 13: Correlations between test examples and model outputs for aggregation operations.