Curate and Generate: A Corpus and Method for Joint Control of Semantics and Style in Neural NLG

Shereen Oraby, Vrindavan Harrison, Abteen Ebrahimi, and Marilyn Walker
Natural Language and Dialog Systems Lab
University of California, Santa Cruz
{soraby,vharriso,aaebrahi,mawalker}@ucsc.edu

Abstract

Neural natural language generation (NNLG) from structured meaning representations has become increasingly popular in recent years. While we have seen progress with generating syntactically correct utterances that preserve semantics, various shortcomings of NNLG systems are clear: new tasks require new training data which is not available or straightforward to acquire, and model outputs are simple and may be dull and repetitive. This paper addresses these two critical challenges in NNLG by: (1) scalably (and at no cost) creating training datasets of parallel meaning representations and reference texts with rich style markup by using data from freely available and naturally descriptive user reviews, and (2) systematically exploring how the style markup enables joint control of semantic and stylistic aspects of neural model output. We present YELPNLG, a corpus of 300,000 rich, parallel meaning representations and highly stylistically varied reference texts spanning different restaurant attributes, and describe a novel methodology that can be scalably reused to generate NLG datasets for other domains. The experiments show that the models control important aspects, including lexical choice of adjectives, output length, and sentiment, allowing the models to successfully hit multiple style targets without sacrificing semantics.

1 Introduction

The increasing popularity of personal assistant dialog systems and the success of end-to-end neural models on problems such as machine translation has lead to a surge of interest around data-to-text neural natural language generation (NNLG). State-of-the-art NNLG models commonly use a sequence-to-sequence framework for end-to-end neural language generation, taking a meaning representation (MR) as input, and generating a natural language (NL) realization as output (Dusek and Jurcícek, 2016; Lampouras and Vlachos, 2016; Mei et al., 2015; Wen et al., 2015b). Table 1 shows some examples of MR to human and system NL realizations from recently popular NNLG datasets.

The real power of NNLG models over traditional statistical generators is their ability to produce natural language output from structured input in a completely data-driven way, without needing hand-crafted rules or templates. However, these models suffer from two critical bottlenecks: (1) a **data bottleneck**, i.e. the lack of large parallel training data of MR to NL, and (2) a **control bottleneck**, i.e. the inability to systematically control important aspects of the generated output to allow for more stylistic variation.

Recent efforts to address the data bottleneck with large corpora for training neural generators have relied almost entirely on high-effort, costly crowdsourcing, asking humans to write references given an input MR. Table 1 shows two recent efforts: the E2E NLG challenge (Novikova et al., 2017a) and the WEBNLG challenge (Gardent et al., 2017), both with an example of an MR, human reference, and system realization. The largest dataset, E2E, consists of 50k instances. Other datasets, such as the Laptop (13k) and TV (7k) product review datasets, are similar but smaller (Wen et al., 2015a,b).

These datasets were created primarily to focus on the task of semantic fidelity, and thus it is very evident from comparing the human and system outputs from each system that the model realizations are less fluent, descriptive, and natural than the human reference. Also, the nature of the domains (restaurant description, Wikipedia infoboxes, and technical product reviews) are not particularly descriptive, exhibiting little variation.

Other work has also focused on the control bottleneck in NNLG, but has zoned in on one particular dimension of style, such as sentiment, length,

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5938–5951
Florence, Italy, July 28 - August 2, 2019.

1 - E2E (Novikova et al., 2017a) **50k - Crowdsourcing (Domain: Restaurant Description)**
MR: name[Blue Spice], eatType[restaurant], food[English], area[riverside], familyFriendly[yes], near[Rainbow Vegetarian Cafe] **Human:** *Situated near the Rainbow Vegetarian Cafe in the riverside area of the city, The Blue Spice restaurant is ideal if you fancy traditional English food whilst out with the kids.* **System:** Blue Spice is a family friendly English restaurant in the riverside area near Rainbow Vegetarian Cafe.
2 - WebNLG (Gardent et al., 2017) **21k - DBPedia and Crowdsourcing (Domain: Wikipedia)**
MR: (Buzz-Aldrin, mission, Apollo-11), (Buzz-Aldrin, birthname, "Edwin Eugene Aldrin Jr."), (Buzz-Aldrin, awards, 20), (Apollo-11, operator, NASA) **Human:** *Buzz Aldrin (born as Edwin Eugene Aldrin Jr) was a crew member for NASA's Apollo 11 and had 20 awards.* **System:** Buzz aldrin, who was born in edwin eugene aldrin jr., was a crew member of the nasa operated apollo 11. he was awarded 20 by nasa.
3 - YelpNLG (this work) **300k - Auto. Extraction (Domain: Restaurant Review)**
MR: (attr=food, val=taco, adj=no-adj, mention=1), (attr=food, val=flour-tortilla, adj=small, mention=1), (attr=food, val=beef, adj=marinated, mention=1), (attr=food, val=sauce, adj=spicy, mention=1) +[sentiment=positive, len=long, first-person=false, exclamation=false] **Human:** *The taco was a small flour tortilla topped with marinated grilled beef, asian slaw and a spicy delicious sauce.* **System:** The taco was a small flour tortilla with marinated beef and a spicy sauce that was a nice touch.

Table 1: A comparison of popular NNLG datasets.

(1/5 star) *I want to curse everyone I know who recommended this craptacular buffet. [...] It's absurdly overpriced at more than $50 a person for dinner. What do you get for that princely sum? Some cold crab legs (it's NOT King Crab, either, despite what others are saying) Shrimp cocktail (several of which weren't even deveined. GROSS. [...])*
(5/5 star) *One of my new fave buffets in Vegas! Very cute interior, and lots of yummy foods! [...] The delicious Fresh, delicious king grab legs!! [...]REALLY yummy desserts! [...] All were grrreat, but that tres leches was ridiculously delicious.*

Table 2: Yelp restaurant reviews for the same business.

or formality (Fan et al., 2017; Hu et al., 2017; Ficler and Goldberg, 2017; Shen et al., 2017; Herzig et al., 2017; Fu et al., 2018; Rao and Tetreault, 2018). However, human language actually involves a constellation of interacting aspects of style, and NNLG models should be able to jointly control these multiple interacting aspects.

In this work, we tackle **both** bottlenecks simultaneously by leveraging masses of freely available, highly descriptive user review data, such as that shown in Table 2. These naturally-occurring examples show a highly positive and highly negative review for the same restaurant, with many examples of rich language and detailed descriptions, such as *"absurdly overpriced"*, and *"ridiculously delicious"*. Given the richness of this type of free, abundant data, we ask: (1) can this freely available data be used for training NNLG models?, and (2) is it possible to exploit the variation in the data to develop models that jointly control multiple interacting aspects of semantics and style?

We address these questions by creating the YELPNLG corpus, consisting of 300k MR to reference pairs for training NNLGs, collected *completely automatically* using freely available data (such as that in Table 2), and off-the-shelf tools.[1] Rather than starting with a meaning representation and collecting human references, we begin with the references (in the form of review sentences), and work backwards – systematically constructing meaning representations for the sentences using dependency parses and rich sets of lexical, syntactic, and sentiment information, including ontological knowledge from DBPedia. This method uniquely exploits existing data which is naturally rich in semantic content, emotion, and varied language. Row 3 of Table 1 shows an example MR from YELPNLG, consisting of relational tuples of attributes, values, adjectives, and order information, as well as sentence-level information including sentiment, length, and pronouns.

Once we have created the YELPNLG corpus, we are in the unique position of being able to explore, for the first time, how varying levels of supervision in the encoding of content, lexical choice, and sentiment can be exploited to control style in NNLG. Our contributions include:

- A new corpus, YELPNLG, larger and more lexically and stylistically varied than existing NLG datasets;
- A method for creating corpora such as YELPNLG, which should be applicable to other domains;
- Experiments on controlling multiple interacting aspects of style with an NNLG while maintaining semantic fidelity, and results using a broad range of evaluation methods;
- The first experiments, to our knowledge, showing that an NNLG can be trained to control lexical choice of adjectives.

We leave a detailed review of prior work to Section 5 where we can compare it with our own.

[1] https://nlds.soe.ucsc.edu/yelpnlg

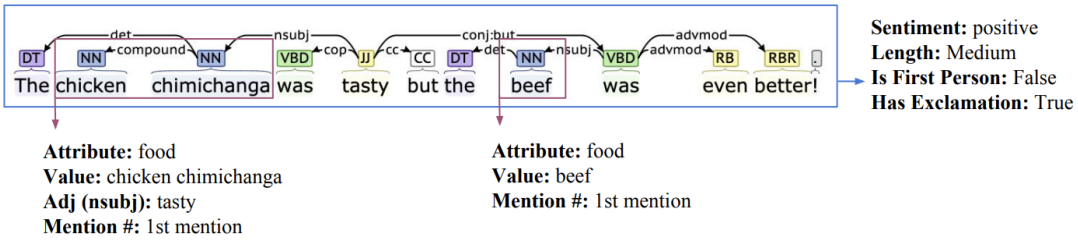

Figure 1: Extracting information from a review sentence parse to create an MR.

2 Creating the YelpNLG Corpus

We begin with reviews from the Yelp challenge dataset,[2] which is publicly available and includes structured information for attributes such as location, ambience, and parking availability for over 150k businesses, with around 4 million reviews in total. We note that this domain and dataset are particularly unique in how naturally descriptive the language used is, as exemplified in Table 2, especially compared to other datasets previously used for NLG in domains such as Wikipedia.

For corpus creation, we must first sample sentences from reviews in such a way as to allow the automatic and reliable construction of MRs using fully automatic tools. To identify restaurant attributes, we use restaurant lexicons from our previous work on template-based NLG (Oraby et al., 2017). The lexicons include five attribute types prevalent in restaurant reviews: *restaurant-type, cuisine, food, service,* and *staff* collected from Wikipedia and DBpedia, including, for example, around 4k for *foods* (e.g. "sushi"), and around 40 for *cuisines* (e.g. "Italian"). We then expand these basic lexicons by adding in attributes for *ambiance* (e.g. "decoration") and *price* (e.g. "cost") using vocabulary items from the E2E generation challenge (Novikova et al., 2017b).

To enforce some semantic constraints and "truth grounding" when selecting sentences without severely limiting variability, we only select sentences that mention particular food values. A pilot analysis of random reviews show that some of the most commonly mentioned foods are meat items, i.e. *"meat", "beef", "chicken", "crab",* and *"steak"*. Beginning with the original set of over 4 million business reviews, we sentence-tokenize them and randomly sample a set of 500,000 sentences from restaurant reviews that mention of at least one of the meat items (spanning around 3k unique restaurants, 170k users, and 340k reviews).

We filter to select sentences that are between 4 and 30 words in length: restricting the length increases the likelihood of a successful parse and reduces noise in the process of automatic MR construction. We parse the sentences using Stanford dependency parser (Chen and Manning, 2014), removing any sentence that is tagged as a fragment. We show a sample sentence parse in Figure 1. We identify all nouns and search for them in the attribute lexicons, constructing *(attribute, value)* tuples if a noun is found in a lexicon, including the full noun compound if applicable, e.g. *(food, chicken-chimichanga)* in Figure 1.[3] Next, for each *(attribute, value)* tuple, we extract all *amod*, *nsubj*, or *compound* relations between a noun value in the lexicons and an adjective using the dependency parse, resulting in *(attribute, value, adjective)* tuples. We add in *"mention order"* into the tuple distinguish values mentioned multiple times in the same reference.

We also collect sentence-level information to encode additional style variables. For *sentiment*, we tag each sentence with the sentiment inherited from the "star rating" of the original review it appears in, binned into one of three values for lower granularity: **1** for low review scores (1-2 stars), **2** for neutral scores (3 star), and **3** for high scores (4-5 stars).[4] To experiment with control of length, we assign a *length* bin of *short* ($\leq$ 10 words), *medium* (10-20 words), and *long* ($\geq$ 20 words). We also include whether the sentence is in first person.

For each sentence, we create 4 MR variations. The simplest variation, BASE, contains only attributes and their values. The +ADJ version adds adjectives, +SENT adds sentiment, and finally the richest MR, +STYLE, adds style information on

[2] https://www.yelp.com/dataset/challenge

[3] Including noun compounds allows us to identify new values that did not exist in our lexicons, thus automatically expanding them.

[4] A pilot experiment comparing this method with Stanford sentiment (Socher et al., 2013) showed that copying down the original review ratings gives more reliable sentiment scores.

1	**The chicken chimichanga was tasty but the beef was even better!** (*attr*=food, *val*=chicken_chimichanga, *adj*=tasty, *mention*=1), (*attr*=food, *val*=beef, *adj*=no_adj, *mention*=1) +[*sentiment*=positive, *len*=medium, *first_person*=false, *exclamation*=true]
2	**Food was pretty good (i had a chicken wrap) but service was crazy slow.** (*attr*=food, *val*=chicken_wrap, *adj*=no_adj, *mention*=1), (*attr*=service, *val*=service, *adj*=slow, *mention*=1) +[*sentiment*=neutral, *len*=medium, *first_person*=true, *exclamation*=false]
3	**The chicken was a bit bland ; i prefer spicy chicken or well seasoned chicken.** (*attr*=food, *val*=chicken, *adj*=bland, *mention*=1), (*attr*=food, *val*=chicken, *adj*=spicy, *mention*=2), (*attr*=food, *val*=chicken, *adj*=seasoned, *mention*=3) +[*sentiment*=neutral, *len*=medium, *first_person*=true, *exclamation*=false]
4	**The beef and chicken kebabs were succulent and worked well with buttered rice, broiled tomatoes and raw onions.** (*attr*=food, *val*=beef_chicken_kebabs, *adj*=succulent, *mention*=1), (*attr*=food, *val*=rice, *adj*=buttered, *mention*=1), (*attr*=food, *val*=tomatoes, *adj*=broiled, *mention*=1), (*attr*=food, *val*=onions, *adj*=raw, *mention*=1) +[*sentiment*=positive, *len*=long, *first_person*=false, *exclamation*=false]

Table 3: Sample sentences and automatically generated MRs from YELPNLG. Note the stylistic variation that is marked up in the +STYLE MRs, especially compared to those in other corpora such as E2E or WEBNLG.

mention order, whether the sentence is first person, and whether it contains an exclamation. Half of the sentences are in first person and around 10% contain an exclamation, and both of these can contribute to controllable generation: previous work has explored the effect of first person sentences on user perceptions of dialog systems (Boyce and Gorin, 1996), and exclamations may be correlated with aspects of a hyperbolic style.

Table 3 shows sample sentences for the richest version of the MR (+STYLE) that we create. In Row 1, we see the MR from the example in Figure 1, showing an example of a NN compound, "chicken chimichanga", with adjective "tasty", and the other food item, "beef", with no retrieved adjective. Row 2 shows an example of a "service" attribute with adjective "slow", in the first person, and neutral sentiment. Note that in this example, the method does not retrieve that the "chicken wrap" is actually described as "good", based on the information available in the parse, but that much of the other information in the sentence is accurately captured. We expect the language model to successfully smooth noise in the training data caused by parser or extraction errors.[5] Row 3 shows an example of the value "chicken" mentioned 3 times, each with different adjectives ("bland", "spicy", and "seasoned"). Row 4 shows an example of 4 foods and very positive sentiment.

2.1 Comparison to Previous Datasets

Table 4 compares YELPNLG to previous work in terms of data size, unique vocab and adjectives, entropy,[6] average reference length (RefLen), and examples of stylistic and structural variation in terms of contrast (markers such as "but" and "although"), and aggregation (e.g. "both" and "also") (Juraska and Walker, 2018), showing how our dataset is much larger and more varied than previous work. We note that the Laptop and E2E datasets (which allow multiple sentences per references) have longer references on average than YelpNLG (where references are always single sentences and have a maximum of 30 words). We are interested in experimenting with longer references, possibly with multiple sentences, in future work.

Figure 2 shows the distribution of MR length, in terms of the number of attribute-value tuples. There is naturally a higher density of shorter MRs, with around 13k instances from the dataset containing around 2.5 attribute-value tuples, but that the MRs go up to 11 tuples in length.

	E2E	LAPTOP	YELPNLG
Train Size	42k	8k	**235k**
Train Vocab	2,786	1,744	**41,337**
Train # Adjs	944	381	**13,097**
Train Entropy	11.59	11.57	**15.25**
Train RefLen	22.4	**26.4**	17.32
% Refs w/ Contrast	5.78%	3.61%	**9.11%**
% Refs w/ Aggreg.	1.64%	2.54%	**6.39%**

Table 4: NLG corpus statistics from E2E (Novikova et al., 2017a), LAPTOP (Wen et al., 2016), and YELPNLG (this work).

2.2 Quality Evaluation

We examine the quality of the MR extraction with a qualitative study evaluating YELPNLG MR to NL

[5] We note that the Stanford dependency parser (Chen and Manning, 2014) has a token-wise labeled attachment score (LAS) of 90.7, but point out that for our MRs we are primarily concerned with capturing NN compounds and adjective-noun relations, which we evaluate in Section 2.2.

[6] We show the formula for entropy in Sec 4 on evaluation.

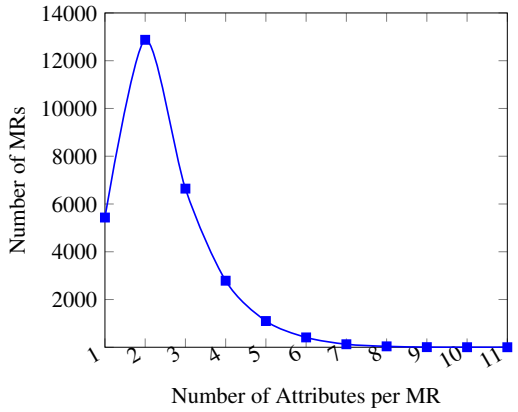

Figure 2: MR distribution in YELPNLG train.

pairs on various dimensions. Specifically, we evaluate **content preservation** (how much of the MR content appears in the NL, specifically, nouns and their corresponding adjectives from our parses), **fluency** (how "natural sounding" the NL is, aiming for both grammatical errors and general fluency), and **sentiment** (what the perceived sentiment of the NL is). We note that we conduct the same study over our NNLG test outputs when we generate data using YELPNLG in Section 4.3.

We randomly sample 200 MRs from the YELPNLG dataset, along with their corresponding NL references, and ask 5 annotators on Mechanical Turk to rate each output on a 5 point Likert scale (where 1 is low and 5 is high for content and fluency, and where 1 is negative and 5 is positive for sentiment). For content and fluency, we compute the average score across all 5 raters for each item, and average those scores to get a final rating for each model, such that higher content and fluency scores are better. We compute sentiment error by converting the judgments into 3 bins to match the Yelp review scores (as we did during MR creation), finding the average rating for all 5 annotators per item, then computing the *difference* between their average score and the true sentiment rating in the reference text (from the original review), such that lower sentiment error is better.

The average ratings for content and fluency are high, at 4.63 and 4.44 out of 5, respectively, meaning that there are few mistakes in marking attribute and value pairs in the NL references, and that the references are also fluent. This is an important check because correct grammar/spelling/punctuation is not a restriction in Yelp reviews. For sentiment, the largest error is 0.58 (out of 3), meaning that the perceived sentiment by raters does not diverge greatly, on average, from the Yelp review sentiment assigned in the MR, and indicates that inheriting sentence sentiment from the review is a reasonable heuristic.

3 Model Design

In the standard RNN encoder-decoder architecture commonly used for machine translation (Sutskever et al., 2014; Bahdanau et al., 2014), the probability of a target sentence $w_{1:T}$ given a source sentence $x_{1:S}$ is modeled as $p(w_{1:T}|x) = \prod_1^T p(w_t|w_{1:t-1}, x)$ (Klein et al., 2018).

In our case, the input is not a natural language source sentence as in traditional machine translation; instead, the input $x_{1:S}$ is a meaning representation, where each token x_n is itself a tuple of attribute and value features, (f_{attr}, f_{val}). Thus, we represent a given input $x_{1:S}$ as a sequence of attribute-value pairs from an input MR. For example, in the case of BASE MR *[(attr=food, val=steak), (attr=food, val=chicken)]*, we would have $x = x_1, x_2$, where x_1=*(f_{attr}=food,f_{val}=steak)*, and x_2=*(f_{attr}=food,f_{val}=chicken)*. The target sequence is a natural language sentence, which in this example might be, *"The steak was extra juicy and the chicken was delicious!"*

Base encoding. During the encoding phase for BASE MRs, the model takes as input the MR as a sequence of attribute-value pairs. We precompute separate vocabularies for attributes and values. MR attributes are represented as vectors and MR values are represented with reduced dimensional embeddings that get updated during training. The attributes and values of the input MR are concatenated to produce a sequence of attribute-value pairs that then is encoded using a multi-layer bidirectional LSTM (Hochreiter and Schmidhuber, 1997).

Additional feature encoding. For the +ADJ, +SENT, and +STYLE MRs, each MR is a longer relational tuple, with additional style feature information to encode, such that an input sequence $x_{1:S} = (f_{attr}, f_{val}, f_{1:N})$, and where each f_n is an additional feature, such as adjective or mention order. Specifically in the case of +STYLE MRs, the additional features may be sentence-level features, such as sentiment, length, or exclamation.

In this case, we enforce additional constraints

on the models for +ADJ, +SENT, and +STYLE, changing the conditional probability computation for $w_{1:T}$ given a source sentence $x_{1:S}$ to $p(w_{1:T}|x) = \prod_1^T p(w_t|w_{1:t-1}, x, f)$, where f is the set of new feature constraints to the model.

We represent these additional features as a vector of additional supervision tokens or side constraints (Sennrich et al., 2016). Thus, we construct a vector for each set of features, and concatenate them to the end of each attribute-value pair, encoding the full sequence as for BASE above.

Target decoding. At each time step of the decoding phase the decoder computes a new decoder hidden state based on the previously predicted word and an attentionally-weighted average of the encoder hidden states. The conditional next-word distribution $p(w_t|w_{1:t-1}, x, f)$ depends on f, the stylistic feature constraints added as supervision. This is produced using the decoder hidden state to compute a distribution over the vocabulary of target side words. The decoder is a unidirectional multi-layer LSTM and attention is calculated as in Luong et al. (2015) using the *general* method of computing attention scores. We present model configurations in Appendix A.

4 Evaluation

To evaluate whether the models effectively hit semantic and stylistic targets, we randomly split the YELPNLG corpus into 80% train (∼235k instances), 10% dev and test (∼30k instances each), and create 4 versions of the corpus: BASE, +ADJ, +SENT, and +STYLE, each with the same split.[7]

Table 5 shows examples of output generated by the models for a given test MR, showing the effects of training models with increasing information. Note that we present the longest version of the MR (that used for the +STYLE model), so the BASE, +ADJ, and +SENT models use the same MR minus the additional information. Row 1 shows an example of partially correct sentiment for BASE, and fully correct sentiment for the rest; +ADJ gets the adjectives right, +SENT is more descriptive, and +STYLE hits all targets. Row 2 gives an example of extra length in +STYLE, *"the meat was so tender and juicy that it melted in your mouth"*. Row 3 shows an example of a negative sentiment target, which is achieved by both the +SENT and +STYLE models, with interesting descriptions such as *"the breakfast pizza was a joke"*, and *"the pizza crust was a little on the bland side"*. We show more +STYLE model outputs in Appendix C.

4.1 Automatic Semantic Evaluation

Machine Translation Metrics. We begin with an automatic evaluation using standard metrics frequently used for machine translation. We use the script provided by the E2E Generation Challenge[8] to compute scores for each of the 4 model test outputs compared to the original Yelp review sentences in the corresponding test set. Rows 1-4 of Table 6 summarize the results for BLEU (n-gram precision), METEOR (n-grams with synonym recall), CIDEr (weighted n-gram cosine similarity), and NIST (weighted n-gram precision), where higher numbers indicate better overlap (shown with the ↑). We note that while these measures are common for machine translation, they are not well-suited to this task, since they are based on n-gram overlap which is not a constraint within the model; we include them for comparative purposes.

From the table, we observe that across all metrics, we see a steady increase as more information is added. Overall, the +STYLE model has the highest scores for all metrics, i.e. +STYLE model outputs are most lexically similar to the references.

Semantic Error Rate. The *types of semantic errors* the models make are more relevant than how well they conform to test references. We calculate average Semantic Error Rate (SER), which is a function of the number of semantic mistakes the model makes (Wen et al., 2015a; Reed et al., 2018). We find counts of two types of common mistakes: deletions, where the model fails to realize a value from the input MR, and repetitions, where the model repeats the same value more than once.[9] Thus, we compute SER per MR as $SER = \frac{D+R}{N}$, where D and R are the number of deletions and repetitions, and the N is the number of tuples in the MR, and average across the test outputs.

[7] Since we randomly split the data, we compute the overlap between train and test for each corpus version, noting that around 14% of test MRs exist in training for the most specific +STYLE version (around 4.3k of the 30k), but that less than 0.5% of the 30k full MR-ref pairs from test exist in train.

[8] https://github.com/tuetschek/e2e-metrics

[9] We note that other types of errors include insertions and substitutions, but we evaluate these through our human evaluation in Sec 4.3 since our large vocabulary size makes identifying them non-trivial.

1	MR	**(food, porridge, no_adj, mention=1), (food, meat, no_adj, mention=1), (food, flavor, rich, mention=1), (food, soup, no_adj, mention=1) +[sentiment=positive, len=long, first_person=false, exclamation=false]**
	BASE	The porridge was good, but the meat lacked flavor and the soup was bland.
	+ADJ	The porridge had a lot of meat in it and the flavor of the soup was rich.
	+SENT	The porridge had a lot of meat in it and the flavor of the soup was rich and delicious.
	+STYLE	The porridge had a good amount of meat and rich flavor, and the soup was cooked perfectly.
2	MR	**(food, gyro_salad, no_adj, mention=1), (food, meat, no_adj, mention=1) +[sentiment=positive, len=long, first_person=true, exclamation=false]**
	BASE	I had the gyro salad and the meat was very good.
	+ADJ	I had the gyro salad and the meat was tender and juicy.
	+SENT	I had the gyro salad and the meat was tender.
	+STYLE	I had the gyro salad and the meat was so tender and juicy that it melted in your mouth.
3	MR	**(food, eggs, no_adj, mention=1), (food, ham_steak, small, mention=1), (food, bacon, chewy, mention=1), (food, breakfast_pizza, no_adj, mention=1) +[sentiment=negative, len=long, first_person=true, exclamation=false]**
	BASE	I had the eggs, ham steak, bacon, and buffalo pizza.
	+ADJ	Eggs, ham steak, chewy bacon, and breakfast pizza.
	+SENT	The eggs were over cooked, the ham steak was small, the bacon was chewy, and the breakfast pizza was a joke.
	+STYLE	I ordered the eggs benedict and the ham steak was small, the bacon was chewy and the pizza crust was a little on the bland side.

Table 5: Sample test MR and corresponding outputs for each model. Note that the MR presented is for +STYLE: the other models all provide less information as described in Section 2.

			BASE	+ADJ	+SENT	+STYLE
1	BLEU	↑	0.126	0.164	0.166	**0.173**
2	METEOR	↑	0.206	0.233	0.234	**0.235**
3	CIDEr	↑	1.300	1.686	1.692	**1.838**
4	NIST	↑	3.840	4.547	4.477	**5.537**
5	Avg SER	↓	**0.053**	0.063	0.064	0.090

Table 6: Automatic semantic evaluation (higher is better for all but SER).

Table 6 presents the average SER rates for each model, where lower rates mean fewer mistakes (indicated by ↓). It is important to note here that we compute errors over value and adjective slots only, since these are the ones that we are able to identify lexically (we cannot identify whether an output makes an error on sentiment in this way, so we measure that with a human evaluation in Section 4.3). This means that the BASE outputs errors are computed over only value slots (since they don't contain adjectives), and the rest of the errors are computed over both value and adjective slots.

Amazingly, overall, Table 6 results show the SER is extremely low, even while achieving a large amount of stylistic variation. Naturally, BASE, with no access to style information, has the best (lowest) SER. But we note that there is not a large increase in SER as more information is added – even for the most difficult setting, +STYLE, the models make an error on less than 10% of the slots in a given MR, on average.

4.2 Automatic Stylistic Evaluation

We compute stylistic metrics to compare the model outputs, with results shown in Table 7.[10] For **vocab**, we find the number of unique words in all outputs for each model. We find the average **sentence length** (SentLen) by counting the number of words, and find the total number of times an **adjective** is used (Row 3) and average number of adjectives per reference for each model (Row 4). We compute Shannon text **entropy** (E) as: $E = -\sum_{x \in V} \frac{f}{t} * log_2(\frac{f}{t})$, where V is the vocab size in all outputs generated by the model, f is the frequency of a term (in this case, a trigram), and t counts the number of terms in all outputs. Finally, we count the instances of **contrast** (e.g. "but" and "although"), and **aggregation** (e.g. "both" and "also"). For all metrics, higher scores indicate more variability (indicated by ↑).

From the table, we see that overall the vocabulary is large, even when compared to the training data for E2E and Laptop, as shown in Table 4. First, we see that the simplest, least constrained BASE model has the largest vocabulary, since it has the most freedom in terms of word choice, while the model with the largest amount of supervision, +STYLE, has the smallest vocab, since we provide it with the most constraints on word choice. For all other metrics, we see that the +STYLE

[10] These measures can be compared to Table 4, which includes similar statistics for the YelpNLG training data.

			BASE	+ADJ	+SENT	+STYLE
1	Vocab	↑	**8,627**	8,283	8,303	7,878
2	SentLen	↑	11.27	11.45	11.30	**13.91**
3	# Adjs	↑	24k	26k	26k	**37k**
4	Adj/Ref	↑	0.82	0.90	0.89	**1.26**
5	Entropy	↑	11.18	11.87	11.93	**11.94**
6	Contrast	↑	1,586	1,000	890	**2,769**
7	Aggreg.	↑	116	103	106	**1,178**

Table 7: Automatic stylistic evaluation metrics (higher is better). Paired t-test BASE vs. +STYLE all $p < 0.05$.

model scores highest: these results are especially interesting when considering that +STYLE has the smallest vocab; even though word choice is constrained with richer style markup, +STYLE is more descriptive on average (more adjectives used), and has the highest entropy (more diverse word collocations). This is also very clear from the significantly higher number of contrast and aggregation operations in the +STYLE outputs.

Language Template Variations. Since our test set consists of 30k MRs, we are able to broadly characterize and quantify the kinds of sentence constructions we get for each set of model outputs. To make generalized sentence templates, we delexicalize each reference in the model outputs, i.e. we replace any food item with a token [FOOD], any service item with [SERVICE], etc. Then, we find the total number of unique templates each model produces, finding that each "more informed" model produces more unique templates: BASE produces 18k, +ADJ produces 22k, +SENT produces 23k, and +STYLE produces 26k unique templates. In other words, given the test set of 30k, +STYLE produces a novel templated output for over 86% of the input MRs.

While it is interesting to note that each "more informed" model produces more unique templates, we also want to characterize *how frequently templates are reused*. Figure 3 shows the number of times each model repeats its top 20 most frequently used templates. For example, the Rank 1 most frequently used template for the BASE model is *"I had the [FOOD] [FOOD]."*, and it is used 550 times (out of the 30k outputs). For +STYLE, the Rank 1 most frequently used template is *"I had the [FOOD] [FOOD] and it was delicious."*, and it is only used 130 times. The number of repetitions decreases as the template rank moves from 1 to 20, and repetition count is always significantly lower for +STYLE, indicating more variation. Examples of frequent templates from the BASE and +STYLE models are are shown in Appendix B.

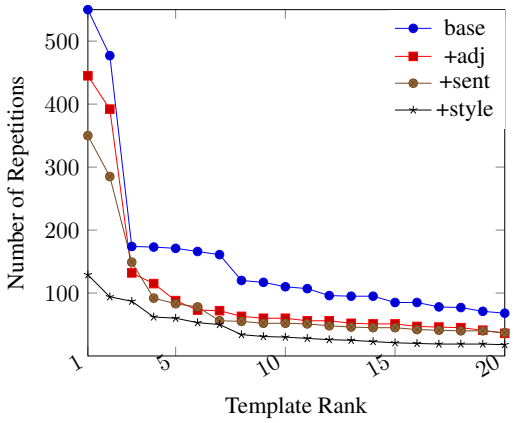

Figure 3: Number of output template repetitions for the 20 most frequent templates (+STYLE has the fewest repetitions, i.e. it is the most varied).

Achieving Other Style Goals. The +STYLE model is the only one with access to first-person, length, and exclamation markup, so we also measure its ability to hit these stylistic goals. The average sentence length for the +STYLE model for LEN=SHORT is 7.06 words, LEN=MED is 13.08, and LEN=LONG is 22.74, closely matching the average lengths of the test references in those cases, i.e. 6.33, 11.05, and 19.03, respectively. The model correctly hits the target 99% of the time for first person (it is asked to produce this for 15k of the 30k test instances), and 100% of the time for exclamation (2k instances require exclamation).

4.3 Human Quality Evaluation

We evaluate output quality using human annotators on Mechanical Turk. As in our corpus quality evaluation from Section 2.2, we randomly sample 200 MRs from the test set, along with the corresponding outputs for each of the 4 models, and ask 5 annotators to rate each output on a 1-5 Likert scale for **content**, **fluency**, and **sentiment** (1 for very negative, 5 for very positive[11]). Table 8 shows the average scores by criteria and model.[12]

For content and fluency, all average ratings are very high, above 4.3 (out of 5). The differences between models are small, but it is interesting

[11] As in Sec 2.2, we scale the sentiment scores into 3 bins to match our Yelp review sentiment.

[12] The average correlation between each annotator's ratings and the average rating for each item is 0.73.

to note that the BASE and +STYLE models are almost tied on fluency (although BASE outputs may appear more fluent due to their comparably shorter length). In the case of sentiment error, the largest error is 0.75 (out of 3), with the smallest sentiment error (0.56) achieved by the +STYLE model. Examination of the outputs reveals that the most common sentiment error is producing a neutral sentence when negative sentiment is specified. This may be due to the lower frequency of negative sentiment in the corpus as well as noise in automatic sentiment annotation.

		BASE	+ADJ	+SENT	+STYLE
Content	↑	4.35*	4.53	4.51	4.49
Fluency	↑	4.43	4.36	4.37	4.41
Sentiment Err	↓	0.75*	0.71*	0.67*	0.56

Table 8: Human quality evaluation (higher is better for content and fluency, lower is better for sentiment error). Paired t-test for each model vs.+STYLE, * is $p < 0.05$.

5 Related Work

Recent efforts on data acquisition for NNLG has relied almost exclusively on crowdsourcing. Novikova et al. (2017a) used pictorial representations of restaurant MRs to elicit 50k varied restaurant descriptions through crowdsourcing. Wen et al. (2015a; 2015b) also create datasets for the restaurant (5k), hotel (5k), laptop (13k), and TV (7k) domains by asking Turkers to write NL realizations for different combinations of input dialog acts in the MR. Work on the WEBNLG challenge has also focused on using existing structured data, such as DBPedia, as input into an NLG (Gardent et al., 2017), where matching NL utterances are also crowdsourced. Other recent work on collecting datasets for dialog modeling also use large-scale crowdsourcing (Budzianowski et al., 2018).

Here, we completely avoid having to crowdsource any data by working in reverse: we begin with naturally occurring user reviews, and automatically construct MRs from them. This allows us to create a novel dataset YELPNLG, the largest existing NLG dataset, with 300k parallel MR to sentence pairs with rich information on attribute, value, description, and mention order, in addition to a set of sentence-level style information, including sentiment, length, and pronouns.

In terms of control mechanisms, very recent work in NNLG has begun to explore using an explicit sentence planning stage and hierarchical structures (Moryossef et al., 2019; Balakrishnan et al., 2019). In our own work, we show how we are able to control various aspects of style with simple supervision within the input MR, without requiring a dedicated sentence planner, and in line with the end-to-end neural generation paradigm.

Previous work has primarily attempted to *individually* control aspects of content preservation and style attributes such as formality and verb tense, sentiment (2017), and personality in different domains such as news and product reviews (Fu et al., 2018), movie reviews (Ficler and Goldberg, 2017; Hu et al., 2017), restaurant descriptions (Oraby et al., 2018), and customer care dialogs (Herzig et al., 2017). To our knowledge, our work is the very first to generate realizations that both express particular semantics *and* exhibit a particular descriptive or lexical style and sentiment. It is also the first work to our knowledge that controls lexical choice in neural generation, a long standing interest of the NLG community (Barzilay and Lee, 2002; Elhadad, 1992; Radev, 1998; Moser and Moore, 1995; Hirschberg, 2008).

6 Conclusions

This paper presents the YelpNLG corpus, a set of 300,000 parallel sentences and MR pairs generated by sampling freely available review sentences that contain attributes of interest, and automatically constructing MRs for them. The dataset is unique in its huge range of stylistic variation and language richness, particularly compared to existing parallel corpora for NLG. We train different models with varying levels of information related to attributes, adjective dependencies, sentiment, and style information, and present a rigorous set of evaluations to quantify the effect of the style markup on the ability of the models to achieve multiple style goals.

For future work, we plan on exploring other models for NLG, and on providing models with a more detailed input representation in order to help preserve more dependency information, as well as to encode more information on syntactic structures we want to realize in the output. We are also interested in including richer, more semantically-grounded information in our MRs, for example using Abstract Meaning Representations (AMRs) (Dorr et al., 1998; Banarescu et al., 2013; Flanigan et al., 2014). Finally, we are interested in reproducing our corpus generation method on various other domains to allow for the creation of numerous useful datasets for the NLG community.

References

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

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained decoding for neural nlg from compositional representations in task-oriented dialogue. *To appear in Proceedings of ACL 19*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Regina Barzilay and Lillian Lee. 2002. Bootstrapping lexical choice via multiple-sequence alignment. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 164–171. Association for Computational Linguistics.

S. Boyce and A. L. Gorin. 1996. User interface issues for natural spoken dialogue systems. In *Proceedings of International Symposium on Spoken Dialogue*, pages 65–68.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. MultiWOZ - a large-scale multi-domain wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750. ACL.

Bonnie J. Dorr, Nizar Habash, and David R. Traum. 1998. A thematic hierarchy for efficient generation from lexical-conceptual structure. In *Proceedings of the Third Conference of the Association for Machine Translation in the Americas on Machine Translation and the Information Soup*, AMTA '98, pages 333–343, London, UK, UK. Springer-Verlag.

Ondrej Dusek and Filip Jurcícek. 2016. A context-aware natural language generator for dialogue systems. *CoRR*, abs/1608.07076.

Michael Elhadad. 1992. *Using Argumentation to Control Lexical Choice: a Functional Unification Implementation*. Ph.D. thesis, Columbia University.

Angela Fan, David Grangier, and Michael Auli. 2017. Controllable abstractive summarization. *CoRR*, abs/1711.05217.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. *CoRR*, abs/1707.02633.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 663–670.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating Training Corpora for NLG Micro-Planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada.

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.

Julia Hirschberg. 2008. Speaking more like you: Lexical, acoustic/prosodic, and discourse entrainment in spoken dialogue systems. In *Proc. of the 8th SIGdial Workshop on Discourse and Dialogue*, page 128.

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

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.

Juraj Juraska and Marilyn Walker. 2018. Characterizing variation in crowd-sourced data for training neural language generators to produce stylistically varied outputs. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 441–450. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander Rush. 2018. Opennmt: Neural machine translation toolkit. In

Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers), pages 177–184. Association for Machine Translation in the Americas.

Gerasimos Lampouras and Andreas Vlachos. 2016. Imitation learning for language generation from unaligned data. In *COLING*, pages 1101–1112. ACL.

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

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2015. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. *CoRR*, abs/1509.00838.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. *CoRR*, abs/1904.03396.

Margaret G. Moser and Johanna Moore. 1995. Investigating cue selection and placement in tutorial discourse. In *ACL 95*, pages 130–137.

Jekaterina Novikova, Ondrej Dušek, and Verena Rieser. 2017a. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Saarbrucken, Germany. ArXiv:1706.09254.

Jekaterina Novikova, Ondrej Dušek, and Verena Rieser. 2017b. The E2E NLG shared task.

Shereen Oraby, Sheideh Homayon, and Marilyn Walker. 2017. Harvesting creative templates for generating stylistically varied restaurant reviews. In *Proceedings of the Workshop on Stylistic Variation*, pages 28–36, Copenhagen, Denmark. Association for Computational Linguistics.

Shereen Oraby, Lena Reed, Shubhangi Tandon, Sharath TS, Stephanie Lukin, and Marilyn Walker. 2018. Controlling personality-based stylistic variation with neural natural language generators. In *Proceedings of the 19th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, page 15321543.

Dragomir R. Radev. 1998. Learning correlations between linguistic indicators and semantic constraints: Reuse of context-dependent descriptions of entities. In *COLING-ACL*, pages 1072–1078.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer.

Lena Reed, Shereen Oraby, and Marilyn Walker. 2018. Can neural generators for dialogue learn sentence planning and discourse structuring? In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 284–295. Association for Computational Linguistics.

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*, pages 35–40. Association for Computational Linguistics.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *NIPS*, pages 6833–6844.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Stroudsburg, PA. 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, Lina Maria Rojas-Barahona, Pei hao Su, David Vandyke, and Steve J. Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *HLT-NAACL*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015a. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-hao Su, David Vandyke, and Steve J. Young. 2015b. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *CoRR*, abs/1508.01745.

Appendix

A Model Configurations

Here we describe final model configurations for the most complex model, +STYLE, after experimenting with different parameter settings. The encoder and decoder are each three layer LSTMs with 600 units. We use Dropout (Srivastava et al., 2014) of 0.3 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 64. We use a learning rate of 1.0 with a decay rate of 0.5 that gets applied after each training epoch starting with the fifth epoch. Gradients are clipped when the absolute value is greater than 5. We tune model hyper-parameters on a development dataset and select the model of lowest perplexity to evaluate on the test dataset. Beam search with three beams is used during inference. MRs are represented using 300 dimensional embeddings. The target side word embeddings are initialized using pre-trained Glove word vectors (Pennington et al., 2014) which get updated during training. Models are trained using lowercased reference texts.

B Repeated Templates from BASE and +STYLE

Table 9 shows the top 10 most repeated templates for the BASE and +STYLE models. Note that "# Reps" indicates the number of times the template is repeated in the test set of 30k instances; the largest number of reps is only 550 for the most frequent BASE model template, only 129 for +STYLE, meaning that the models mostly generate novel outputs for each test instance.

# Reps	BASE Templates
550	*i had the [FOOD] [FOOD].*
477	*i had the [FOOD] and [FOOD].*
174	*i had the [FOOD] [FOOD] [FOOD].*
173	*the [FOOD] [FOOD] was good.*
171	*the [FOOD] and [FOOD] were good.*
166	*the [FOOD] was tender and the [FOOD] was delicious.*
161	*i had the [FOOD] fried [FOOD].*
120	*the [FOOD] [FOOD] was very good.*
117	*the [FOOD] was good but the [FOOD] was a little dry.*
	+STYLE Templates
129	*i had the [FOOD] [FOOD] and it was delicious.*
94	*had the [FOOD] and [FOOD] [FOOD] plate.*
87	*the [FOOD] and [FOOD] were cooked to perfection.*
62	*i had the [FOOD] [FOOD] and it was good.*
60	*i had the [FOOD] [FOOD].*
53	*i had the [FOOD] and my husband had the [FOOD].*
50	*i had the [FOOD] and [FOOD] and it was delicious.*
34	*the [FOOD] and [FOOD] skewers were the only things that were good.*
31	*i had the [FOOD] [FOOD] [FOOD] and it was delicious.*

Table 9: Sample of 10 "most repeated" templates from BASE and +STYLE.

C Sample Model Outputs for +STYLE

Table 10 shows examples outputs from the +STYLE model, with specific examples of style through different forms of personal pronoun use, contrast, aggregation, and hyperbole in Tables 11-14.

1	*(attr*=food, *val*=meat, *adj*=chewy, *mention*=1), (*attr*=food, *val*=sauce, *adj*=no-adj, *mention*=1), +[*sentiment*=negative, *len*=medium, *first-person*=false, *exclamation*=false] **The meat was chewy and the sauce had no taste.**
2	*(attr*=food, *val*=artichokes, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=beef-carpaccio, *adj*=no-adj, *mention*=1), +[*sentiment*=positive, *len*=long, *first-person*=true, *exclamation*=false] **We started with the artichokes and beef carpaccio , which were the highlights of the meal .**
3	*(attr*=staff, *val*=waitress, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=meat-tips, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=ribs, *adj*=no-adj, *mention*=1), +[*sentiment*=neutral, *len*=long, *first-person*=true, *exclamation*=false] **The waitress came back and told us that they were out of the chicken meat tips and ribs .**
4	*(attr*=food, *val*=chicken-lollipops, *adj*=good, *mention*=1), *(attr*=food, *val*=ambiance, *adj*=nice, *mention*=1), +[*sentiment*=positive, *len*=medium, *first-person*=false, *exclamation*=false] **The chicken lollipops were really good , nice ambience .**
5	*(attr*=food, *val*=meat, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=sausage, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=deli-meats, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=cheeses, *adj*=no-adj, *mention*=1), *(attr*=price, *val*=prices, *adj*=good, *mention*=1), +[*sentiment*=positive, *len*=medium, *first-person*=false, *exclamation*=false] **Geat selection of meat , sausage , deli meats , cheeses , and good prices .**
6	*(attr*=food, *val*=beef-chili, *adj*=amazing, *mention*=1), *(attr*=food, *val*=onion, *adj*=carmalized, *mention*=1), +[*sentiment*=positive, *len*=long, *first-person*=true, *exclamation*=false] **The beef chili was amazing , and i loved the caramelized onions that came with it .**
7	*(attr*=food, *val*=eggs, *adj*=runny, *mention*=1), *(attr*=food, *val*=crab-legs, *adj*=open, *mention*=1), +[*sentiment*=neutral, *len*=long, *first-person*=true, *exclamation*=false] **The eggs were runny , and the open faced crab legs were a little too much for my taste .**
8	*(attr*=food, *val*=chicken-salad, *adj*=grilled, *mention*=1), *(attr*=food, *val*=chicken, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=spices, *adj*=right, *mention*=1), *(attr*=food, *val*=salad, *adj*=fresh, *mention*=1), +[*sentiment*=positive, *len*=long, *first-person*=true, *exclamation*=false] **I had the grilled chicken salad , the chicken was tender and the spices and fresh salad were just right .**

Table 10: Sample test outputs from Model +STYLE.

1	*(attr*=food, *val*=fish-meat, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=horse-radish-sauce, *adj*=no-adj, *mention*=1), +[*sentiment*=positive, *len*=long, *first-person*=true, *exclamation*=false] **I had the fish meat and it was very good, and my husband had the chicken horse-radish-sauce which he loved.**
2	*(attr*=food, *val*=beef, *adj*=no-adj, *mention*=1), *(attr*=restaurant, *val*=restaurant, *adj*=nice, *mention*=1), *(attr*=staff, *val*=waiter, *adj*=friendly, *mention*=1), +[*sentiment*=positive, *len*=long, *first-person*=true, *exclamation*=false] **The beef was tender, the restaurant was nice, and the waiter was friendly and helpful to us.**
3	*(attr*=food, *val*=lobster, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=crab-legs, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=mussels, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=clams, *adj*=no-adj, *mention*=1), +[*sentiment*=positive, *len*=medium, *first-person*=true, *exclamation*=false] **We had lobster, crab legs, mussels and clams.**
4	*(attr*=food, *val*=crab-soup, *adj*=no-adj, *mention*=1), +[*sentiment*=negative, *len*=short, *first-person*=false, *exclamation*=false] **She had the crab soup.**
5	*(attr*=staff, *val*=host, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=steak, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=lobster, *adj*=no-adj, *mention*=1), +[*sentiment*=positive, *len*=long, *first-person*=false, *exclamation*=false] **The host came out with the steak and lobster, and he said it was very good .**

Table 11: Examples of different pronouns from Model +STYLE.

1	*(attr*=food, *val*=kids-chicken-fingers, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=chicken, *adj*=actual, *mention*=1), *(attr*=food, *val*=chicken, *adj*=little, *mention*=2), +[*sentiment*=positive, *len*=long, *first-person*=false, *exclamation*=false] **The kids chicken fingers are made with actual chicken, but the chicken is a little on the dry side.**
2	*(attr*=food, *val*=nachos, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=chicken, *adj*=no-adj, *mention*=1), +[*sentiment*=negative, *len*=long, *first-person*=true, *exclamation*=false] **I ordered the nachos with chicken, and they were pretty good, but nothing to write home about.**
3	*(attr*=food, *val*=chicken-tenders, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=chicken-nuggets, *adj*=no-adj, *mention*=1), +[*sentiment*=neutral, *len*=long, *first-person*=true, *exclamation*=false] **The chicken tenders and chicken nuggets were the only things that were good, but nothing special.**
4	*(attr*=food, *val*=rice, *adj*=good, *mention*=1), *(attr*=food, *val*=meat, *adj*=no-adj, *mention*=1), +[*sentiment*=neutral, *len*=long, *first-person*=true, *exclamation*=false] **The rice was good, but i wish there was more meat in the dish.**

Table 12: Examples of contrast from Model +STYLE.

1	*(attr*=food, *val*=meat, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=sausage, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=deli-meats, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=cheeses, *adj*=no-adj, *mention*=1), *(attr*=price, *val*=prices, *adj*=good, *mention*=1), +[*sentiment*=positive, *len*=medium, *first-person*=false, *exclamation*=false] **Great selection of meat, sausage, deli meats, cheeses, and good prices.**
2	*(attr*=food, *val*=tofu, *adj*=fried, *mention*=1), *(attr*=food, *val*=lemongrass-chicken, *adj*=aforementioned, *mention*=1), +[*sentiment*=neutral, *len*=long, *first-person*=true, *exclamation*=false] **I had the fried tofu and my husband had the lemongrass chicken, both of which were very good.**
3	*(attr*=food, *val*=burgers, *adj*=different, *mention*=1), *(attr*=food, *val*=chicken-club, *adj*=grilled, *mention*=1), +[*sentiment*=positive, *len*=long, *first-person*=true, *exclamation*=false] **We ordered two different burgers and a grilled chicken club, both of which were delicious.**
4	*(attr*=food, *val*=octopus, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=salmon, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=tuna, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=crab, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=squid, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=shrimp, *adj*=no-adj, *mention*=1), +[*sentiment*=positive, *len*=long, *first-person*=false, *exclamation*=true] **Octopus, salmon, tuna, crab, squid, shrimp, etc... all of it was delicious !**

Table 13: Examples of aggregation from Model +STYLE.

1	*(attr*=food, *val*=meat, *adj*=spectacular, *mention*=1), *(attr*=food, *val*=sauces, *adj*=no-adj, *mention*=1), +[*sentiment*=positive, *len*=medium, *first-person*=false, *exclamation*=false] **The meat was spectacular and the sauces were to die for.**
2	*(attr*=food, *val*=maine-lobster, *adj*=heavenly, *mention*=1), *(attr*=food, *val*=crab-bisque, *adj*=no-adj, *mention*=1), +[*sentiment*=positive, *len*=long, *first-person*=false, *exclamation*=false] **The lobster claw was heavenly, and the crab bisque was a nice touch, but not overpowering.**
3	*(attr*=food, *val*=meat-sauce-spaghetti, *adj*=no-adj, *mention*=1), *(attr*=food, *val*=milk-tea, *adj*=cold, *mention*=1), +[*sentiment*=positive, *len*=long, *first-person*=true, *exclamation*=false] **I had the chicken meat sauce spaghetti and it was very good and the cold milk tea was the best i have ever had.**
4	*(attr*=food, *val*=seafood, *adj*=fresh, *mention*=1), *(attr*=food, *val*=chicken, *adj*=fried, *mention*=1), *(attr*=food, *val*=bread-pudding, *adj*=phenomenal, *mention*=1), +[*sentiment*=positive, *len*=long, *first-person*=false, *exclamation*=false] **The seafood was fresh, the fried chicken was great, and the bread pudding was phenomenal.**

Table 14: Examples of hyperbole from Model +STYLE.