UNIVERSITY OF CALIFORNIA
SANTA CRUZ

STYLISTIC CONTROL FOR NEURAL NATURAL LANGUAGE GENERATION

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Shereen M. Oraby

June 2019

The Dissertation of Shereen M. Oraby
is approved:

Professor Marilyn Walker, Chair

Professor Snigdha Chaturvedi

Dr. Dilek Hakkani-Tur

Lori Kletzer
Vice Provost and Dean of Graduate Studies

Contents

List of Figures

List of Tables

Abstract

Stylistic Control for Neural Natural Language Generation

by

Shereen M. Oraby

Neural models for generating text from structured representations of meaning have recently gained popularity in the natural language generation (NLG) community. Instead of using a traditional NLG pipeline involving separate modules for sentence planning and surface realization, neural models combine these steps into a single end-to-end framework. This new paradigm allows for low effort data-driven generation, but makes it very unclear how to control model output and produce the required semantics with the desired syntactic or stylistic constructions for a given application.

This thesis takes on the task of learning to control neural natural language generation systems, with the goal of producing natural language outputs that are both semantically correct and stylistically varied. We tackle three critical bottlenecks of neural NLG: how to introduce a mechanism to produce style with neural generators, how to systematically acquire massive amounts of data required for training them, and how to jointly control semantic and stylistic choices, to allow for more diverse model outputs. We address the style bottleneck by experimenting with different methods for supervision in neural models with a synthetic dataset that we build (PersonageNLG), showing that we can produce diverse sentence planning operations in our model outputs. We address the data bottleneck by using freely available review data to create a massive, highly descriptive, and stylistically diverse corpus for training neural generators (YelpNLG), instead of relying on crowdsourcing. We address the control bottleneck by constructing stylistically rich meaning representations derived from review text based on parse information and freely-available ontologies, providing different forms of supervision to our neural models, and allowing us to produce outputs exhibiting a rich array of stylistic variation from semantically-grounded inputs.

We show that by controlling the nature of our input data and how it is represented in our models, we can control a model's ability to produce a required style without sacrificing its ability to produce fluent outputs that express the required content. Our data and experiments introduce novel methods for producing stylistic variation within a neural natural language generation pipeline, and are generalizable to new domains and style choices.

To my family.

Acknowledgments

My PhD path has not been one that I have walked alone, and I would like to thank the many people that have guided me steadily along the way.

My greatest thanks go to my advisor, Lyn Walker, for her genuinely tireless support throughout my time as a PhD student. I have seen first-hand that she is truly as enthusiastic about language and research as she is brilliant: and I must say that her enthusiasm is contagious. She has pushed me to be more critical of my findings, more principled in my approaches, and above all, to remain curious. She has been, and will continue to be, an inspiration to me both in and out of the research lab.

To Ellen Riloff, who I was so lucky to work with closely in my early PhD days. She helped me come to my own as a researcher: asking me questions like "What is *your* intuition?" in my first few weeks as a PhD student, when I was unsure that I had anything of the sort. In her own way, Ellen taught me to have confidence and pride in my ideas, and to be excited to share them with the world.

To my advancement committee, Lise Getoor, Lyn Walker, Steve Whittaker, Ellen Riloff, and Dilek Hakkani-Tur, whose insights on my early work helped me successfully define the research goals for my next phase of work. And to my defense committee, Lyn Walker, Snigdha Chaturvedi, and Dilek Hakkani-Tur, for taking the time to evaluate my final dissertation and providing me with valuable feedback.

To my labmates, who have most closely shared the thrilling ups and frustrating, but temporary, downs of PhD life with me: Elahe, Amita, Lena, Jiaqi, Geetanjali, Stephanie, Chao, Wen, Harrison, Kevin, Jurik, Brian, and Abteen. I have loved both working with you, and getting to know you so much better on our group conference trips, exploring new places and learning about how similar we really are. Our lab has been a wonderful environment to grow, learn, and best of all, make dear friends.

And of course, to my family and friends, both here and back home in Egypt, who have patiently stood by me on this journey, and lovingly pushed me through. And finally, to Hesham, who has always believed in me unwaveringly, and in turn helped me truly believe in myself. Most of all, *“You make life FUN!”*

Chapter 1

Introduction

"Style is on the surface level, very obviously detectable as the choices between items in a vocabulary, between types of syntactical constructions, between the various ways a text can be woven from the material it is made of."

– J. Karlgren [Karlgren, 2004]

1.1 Overview

Natural Language Generation (NLG) systems are tasked with converting input content and a communicative goal into natural language, to be consumed by users. Since we as humans express our communicative goals with a constellation of interacting aspects of semantics and style, the natural language output of any *effective* NLG system should ideally be able to simulate human language choices. In practical terms, this means that in order to be used within the context of a dialog system, for example, an NLG module must produce natural language that not only expresses the required content and goal, but is also fluent, natural, and interesting to the user.

Table 1.1 shows samples of structured representations of content, commonly referred to as a Meaning Representation (MR), and the corresponding Natural Language (NL) realizations. An MR does not have a rigid form, but traditionally includes a communicative goal (e.g. to *"inform"* or *"recommend"*), and consists of a set of attributes (e.g. *"name"* and *"eatType"* in MR 1) and their corresponding values. The NL output of an NLG system is expected to realize each value within the MR. Two NL realizations are shown for each MR: one written by a human (when prompted with the MR), and one generated by a Neural Natural Language Generation (NNLG) system, currently the state-of-the-art for NLG [Dusek and Jurcícek, 2016].

MR 1	inform(name[Blue Spice], eatType[pub], customerRating[average], near[Burger King])
human	*The Blue Spice pub located near Burger King has been rated average by customers.*
nnlg	*Blue Spice is a pub near Burger King with an average customer rating.*
MR 2	inform(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.*
nnlg	*Blue Spice is a family friendly English restaurant in the riverside area near Rainbow Vegetarian Cafe.*

Table 1.1: Human vs. NNLG realizations for two MRs in the restaurant domain.

The task of NNLG has received an enormous surge of interest within the NLG community, fueled by the successful application of neural models on related tasks such as machine translation [Sutskever et al., 2014, Gasic et al., 2017]. The neural approach to NLG promises to simplify the process of producing high quality natural language in any domain by relying on the neural architecture to automatically learn how to map required input content to output natural language realizations, only requiring parallel corpora of MR to NL as input. This means that NNLG seeks to eliminate the need for intermediate representations or alignment, as is necessary for traditional Statistical Natural Language Generation (SNLG) systems [Reiter and Dale, 2000, Rambow et al., 2001, Stent, 2002, Stent and Molina, 2009]. NNLG also emphasizes the benefits of End-to-End (E2E) training, with a great deal of recent work explicitly claiming that NNLG architectures learn to do **both** sentence planning and surface realization jointly [Dusek and Jurcícek, 2016, Lampouras and Vlachos, 2016, Mei et al., 2016, Wen et al., 2015, Nayak et al., 2017], where they were traditionally two separate pipelined stages in SNLG. We describe the differences between these architectures in detail in Chapter 2.

Despite the prevalence of such claims of low-effort, fully data-driven generation from NNLG systems, it is unclear from recent work that the E2E architecture employed by these systems is in fact capable of producing *controllable* sentence planning as was possible with SNLG systems. The problem is exemplified in the NNLG realizations in Table 1.1, where we see simple, rigid constructions that attempt to include the required content items as briefly as possible, smoothing out any stylistically varied language that is not essential for conveying meaning. There are notable differences in length, word choice, and descriptiveness when compared to the sample human realizations. It is commonly suggested that because NNLG models learn to map input to output in a completely data-driven way, a neural model employs a "frequentist" approach: learning only the simplest and most prevalent way to realize the required content from training.

The reality is that human language involves a constellation of interacting

5/5 STARS	One of my new fave buffets in Vegas! Very cute interior, and lots of yummy foods! [...] * Fresh, delicious king crab legs!! * Tons of shrimp * Salmon pizza - Delicious, came with capers. Thought I'd only eat half, but couldn't help myself from eating the whole thing - best Vegas buffet pizza I've had. * This super thick cut, fatty, tender bacon with mustard grain on the side. Forgot what it was called but that stuff changed my life. I await the day we are reunited.... * REALLY yummy desserts! Sampled a cannoli, sprinkles iced rice krispie treat, tres leches and gelato. All were grrreat, but that tres leches was ridiculously delicious. All of the items out for dinner sounded equally fantastic and tempting, but I unfortunately could not eat any more than the crab legs. Can't wait to come back and try all the dinner items + the supposedly prime rib I didn't have room for!!
1/5 STARS	I want to curse everyone I know who recommended this craptacular buffet. I don't even know where to start. [...] 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. [...]) * A bunch of other unremarkable crap you can find at any other buffet on the Strip. The prime rib was decent, however, and there was some really good mushroom risotto. But those two things in no way justify the price or the wait or the raves this place gets. The only thing that could have made our experience worse is if we had coughed up the extra $12 to skip the line. Now, who do I see to get my $50 and my 60 minutes back?

Table 1.2: Descriptive Yelp restaurant reviews for the same restaurant.

aspects of style. Consider, for example, the restaurant reviews shown in Table 1.2, which show an example of a highly-positive and highly-negative description of the same restaurant. In this case, the data is written organically by users, seeking to *"inform"* as was the case for the generated restaurant descriptions from Table 1.1, but this time including many examples of rich language and detailed descriptions, such as *"absurdly overpriced"*, and *"ridiculously delicious"*. We also see examples of figurative language and hyperbole (*"that stuff changed my life"*), humor and rhetorical questions (*"Now, who do I see to get my $50 and my 60 minutes back?"*), sarcasm

(*"What do you get for that princely sum?"*), and many instances of emphasis through capitalization and elongation (e.g. *"REALLY"*, *"GROSS"*, *"grrreat"*) [Oraby et al., 2017]. These examples of rich language are absent even in the human-generated realizations from Table 1.1, which were authored as part of a writing task, given the input MR, rather than being motivated by real-life experiences and emotions.

Thus, while NNLG models promise a low-cost, data driven method for generation, much of the creative and descriptive elements that make language exciting are lost in translation. The state-of-the-art in NNLG is limited in its ability to generate language that resembles the richness and complexity of human expression. We categorize the limitations of NNLG systems here into three critical bottlenecks, framed as research questions:

1. **Style:** *Can we develop a supervision mechanism to produce style in NNLG models?*
2. **Data:** *Can we create sufficiently large and varied datasets to train NNLGs?*
3. **Control:** *Can we jointly control multiple interacting aspects of style with NNLGs?*

The objective of this thesis is to address these bottlenecks through controllable NNLG. We seek to develop NNLG systems that can produce output that both satisfies the required semantics as defined by an input MR, *and* simultaneously includes interesting and diverse structural and stylistic constructions, such as those in Table 1.2.

We approach the task of generating structurally and stylistically varied outputs in an NNLG by first exploring methods to add supervision to the standard NNLG pipeline to allow it to produce stylistic variation, through a series of experiments with a synthetic dataset that we create. Next, armed with a proof of concept method for controllable NNLG, we develop a novel method for designing a corpus for large-scale NNLG using freely available data in the restaurant domain, such as that exemplified in Table 1.2, as a way to maximize the amount of stylistic variation we see in training, as opposed to using synthetically generated data or data elicited from the crowd as in Table 1.1.

Finally, we present a set of experiments showing how we can jointly control multiple interacting aspects of style in our outputs using our control method and new large-scale dataset, including lexical word choice, sentiment, and length. We note that while our main interest in this thesis is stylistic variation, it is only useful to introduce style to generated outputs if we are simultaneously able to preserve semantics, so we consistently evaluate our methods using both stylistic and semantic measures.

In this chapter, we describe each task aimed at addressing the NNLG bottlenecks, and provide a summary of our contributions.

1.2 Towards Controllable Style in Neural NLG

In this section, we describe our journey towards controllable NLG in this thesis, beginning with a proof-of-concept, followed by our pursuit of data and methods for scalable, controllable neural models.

1.2.1 First Steps towards Producing Style in NNLG

The E2E paradigm employed by NNLG systems, which we describe in detail in Chapter 3, means that there is no clear distinction between sentence planning and surface realization, which are traditionally two separate steps in SNLG. Since there is no separation of tasks, it is unclear how or where to introduce supervision to help guide the generator to make stylistically varied language choices without sacrificing semantics.

As a result, NNLG models are notorious for making semantic errors. Most of the work to date in NNLG has focused almost exclusively on controlling the semantic correctness of outputs, which reduces the job of the sentence planner to making sure that every item in the input MR appears in the output at least once (deletions), that it only appears one time (repetitions) and that content that was not in the MR does not appear in the output (insertions/substitutions). Methods for semantic control include tuning objective functions, penalization methods, and output re-ranking [Dušek and Jurcicek, 2015, Dusek and Jurcícek, 2016, Juraska et al., 2018].

While there has been some success in recent work on improving semantic correctness, this oversimplified view of the role of the sentence planner means that outputs generated by state-of-the-art NNLG are notably lacking in structural and stylistic diversity, and there is little experimentation on effective mechanisms to date to induce the types of variation that were possible in the traditional SNLG pipeline [Dusek et al., 2019]. In addition, the datasets currently used for NNLG, such as the data from the E2E Challenge dataset [Novikova et al., 2017b], shown in Table 1.1, are primarily crowdsourced and mostly designed to train NNLGs to correctly reproduce semantics, and in this setting eliciting realizations that are *also* structurally and stylistically varied realizations from crowd workers is an arduous task [Novikova et al., 2017b].

The first goal of this thesis is to test, for the first time, whether we can systematically introduce some simple, controllable stylistic choices into an NNLG pipeline, *without* sacrificing semantic fidelity, and whether we can measure how well resultant models produce outputs that adhere to the required stylistic choices. To do this, we need: (1) training data where we *know* both semantic and stylistic properties for each instance, and (2) a mechanism for introducing some stylistic supervision to the model, dictating the choices the model should make.

We begin with the E2E challenge corpus, which is a parallel corpus of 50k MR to NL instances in the restaurant domain, acquired through a massive crowdsourcing effort [Novikova et al., 2017c]. This dataset satisfies half of our training data requirement: we know the semantics associated with each instance, but we need a way to introduce some simple stylistic variation into the instances to allow our models to learn different ways of expressing the same content.

To create an experimental setting for controllable NNLG, we utilize the PERSONAGE statistical generator, which takes as input a representation of content, and outputs different ways to express that content, based on a set of rules defining Big-Five personalities [Mairesse and Walker, 2010]. These rules govern decisions such as word choice, pragmatic marker insertion, and aggregation operations, which we describe in detail in Chapter 4.

We sample a set of unique MRs from the E2E Challenge training set, and using PERSONAGE, we create 5 variations for each MR, each one based on a different personality. Table 1.3 shows examples of how a single MR is expressed with five different personalities: *agreeable, disagreeable, conscientious, unconscientious,* and *extravert*. In this way, we synthetically design the PERSONAGENLG corpus: a set of

88k MR to NL utterances in five different personality styles. This corpus provides us with a controlled environment for testing whether an NNLG model can learn to produce both the required content *and* style for a given instance, where personality is a proxy for a multitude of different style choices. While these training realizations are clearly not as natural as the human-written realizations from Table 1.2, this setting provides an environment for testing controllable NNLG, where style choices are finite and defined by parametrized rules.

MR	NAME[FITZBILLIES], EATTYPE[PUB], FOOD[ITALIAN], PRICERANGE[MODERATE], CUSTOMERRATING[DECENT], AREA[RIVERSIDE], FAMILYFRIENDLY[NO],NEAR["THE SORRENTO"]
AGREEABLE	Let's see what we can find on Fitzbillies. I see, well it is a pub with a decent rating, also it is an Italian restaurant in riverside and moderately priced near The Sorrento, also it isn't family friendly, you see?
DISAGREEABLE	I mean, everybody knows that moderately priced Fitzbillies is in riverside with a decent rating. It's near The Sorrento. It isn't family friendly. It is an Italian place. It is a pub.
CONSCIENTIOUS	Let's see what we can find on Fitzbillies. I see, well it is a pub with a decent rating, it isn't kid friendly and it's moderately priced near The Sorrento and an Italian restaurant in riverside.
UNCONSCIENTIOUS	Oh god yeah, I don't know. Fitzbillies is a pub with a decent rating, also it is moderately priced near The Sorrento and an Italian place in riverside and it isn't kid friendly.
EXTRAVERT	Basically, Fitzbillies is an Italian place near The Sorrento and actually moderately priced in riverside, it has a decent rating, it isn't kid friendly and it's a pub, you know.

Table 1.3: Sample realizations for different personalities, given the same MR.

Given the PERSONAGENLG corpus, we experiment with the second requirement for controllable NNLG: a mechanism for introducing stylistic supervision to the model. We develop two different models with different levels of supervision, which we describe in detail in Chapter 4. For the first model, MODEL_TOKEN, we experiment with using a single token to identify which personality style to produce given an input MR. This method is inspired by the use of a single language token for machine translation [Johnson et al., 2016]. For the second model, MODEL_CONTEXT, we experiment with a more detailed form of supervision, where we provide more detailed style parameters from PERSONAGE to the model through an architectural change change, dictating more explicitly what choices the model should make, such as whether to include a particular hedge or pragmatic marker.

We compare each supervision method to a vanilla model that does not use any style encoding. We find that while the vanilla model makes the fewest semantic errors, the outputs loses any distinctive stylistic variation. With MODEL_CONTEXT, however, we are able to achieve our goal: we can *both* produce stylistically varied outputs that correlate with the required personalities, *and* preserve semantic fidelity with notably few errors [Oraby et al., 2018b, Oraby et al., 2018a].

1.2.2 Tackling the Data Bottleneck

Given our success in providing standard NNLG models with some supervision dictating style choices in our synthetic PERSONAGENLG environment, we now set out to explore if we can generate more natural, varied language.

We note that in general, any NLG system is inherently limited by the data it is trained on; if models are not provided with rich examples of stylistically diverse data as input, we cannot expect them to generate them as output, and particular stylistic operations may be very desirable based on the application at hand. For example, if a system is able to control sentence planning operations such as sentence scoping, this affects the complexity of the sentences that compose an output, allowing the generator to produce simpler sentences when desired that might be easier for particular users to understand. Discourse structuring is often critical in persuasive settings such as recommending restaurants, hotels or travel options [Scott and de Souza, 1990, Moore and Paris, 1993], in order to express discourse relations that hold between content items [Stent et al., 2002]. Previous work has explored the effect of first person sentences on user perceptions of dialog systems [Boyce and Gorin, 1996]. Pragmatic variation, such as the use of different pragmatic markers or discourse cues, and lexical choice, or which words to use to express concepts, also have a clear effect on the perceived style of the output.

Recent attempts to create datasets for NNLG involve extensive use of crowdsourcing, where humans are shown some form of MR to provide semantic grounding, and asked to write an NL realization that includes all of the required content items [Novikova et al., 2016, Novikova et al., 2017c, Wen et al., 2015, Gardent et al., 2017a]. Rows 1-3 of Table 1.4 show examples of MR to NL pairs from these datasets, which come from different domains. While crowdsourcing methods do a good job at ensuring that all content items are expressed, they suffer from serious limitations: crowdsourcing (especially large datasets) is costly and time-consuming, and crowd workers may not be motivated to vary their lexical choices or the way in which they write their realizations, resulting in dull and repetitive training data.

In our attempt to find new and better sources of data for training NNLG, we are naturally drawn back to the tantalizing richness of the user reviews from Table 1.2. This type of data is full of all of sorts of charged language and excitingly complex style choices, grounded in peoples' experiences, as opposed to elicited through a carefully designed crowdsourcing task. We ask: *can we use this rich, abundant, freely available data to train NNLGs?*

Since we are primarily concerned with data-to-text generation from a structured MR to an NL to train NNLGs, in order to use this data in any meaningful way, we must first find a way to convert a given NL sentence from a review into a structured MR that represents the sentence's meaning. We present our method to systematically "retrofit" an MR from an NL using entirely off-the-shelf tools, including part-of-speech and dependency information from the sentence parse, and entity-type information from open-source ontologies such as DBPedia.

Using this method, we thus build the YelpNLG corpus, a set of 300k MR to NL realizations created using off-the-shelf tools and freely available data [Oraby et al., 2017, Oraby et al., 2019]. We describe our methods for creating similar corpora for NLG in detail in Chapter 5. Row 4 of Table 1.4 shows an example from our YelpNLG training corpus, showing the types of rich information included in our MRs, as compared to those of popular datasets for NLG. Our YelpNLG MRs not only include attribute-value pairs as in other datasets, but also use a relational tuple format to group together dependency relations for values, e.g. *(food, brioche-bun, yummy)*. We also add additional information describing stylistic features of the NL, characterizing sentiment, length, personal pronouns, and exclamations.

YelpNLG is also significantly larger and more stylistically diverse than existing datasets, which we demonstrate empirically in Chapter 5. As shown in Table 1.4, which compares YelpNLG to other existing datasets, it is over 5 times as large as the E2E dataset [Novikova et al., 2017c], with around 235k training instances, and has a vocabulary of 41k unique words, more than 5 times as large as WebNLG [Gardent et al., 2017a].

Corpus Description	Sample Data	Size	Vocab
1 - LAPTOP Product Descriptions Crowdsourcing [Gardent et al., 2017b]	**MR:** inform(name=satellite eurus 65; type=laptop; memory=4 gb; driverange=medium; isforbusinesscomputing=false) **Human:** *The satellite eurus 65 is a laptop designed for home use with 4 gb of memory and a medium sized hard drive*	8k	1.7k
2 - WEBNLG Wikipedia DBPedia + crowdsourcing [Gardent et al., 2017b]	**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.*	25k	8k
3 - E2E Restaurant Descriptions Crowdsourcing [Novikova et al., 2017b]	**MR:** name[Blue Spice], eatType[coffee shop], customer rating[average], near[Burger King] **Human:** *The Blue Spice coffee shop near Burger King has good customer ratings with excellent food and service, with a lovely atmosphere.*	42k	2.7k
4 - YELPNLG **Restaurant Reviews** **Automatic Extraction** **([Oraby et al., 2017, Oraby et al., 2019])**	**MR:** (food, chicken-sandwich, fried, mention=1), (food, fries, french, mention=1), (food, chicken, no-adj, mention=1), (food, brioche-bun, yummy, mention=1) +[sentiment=positive, len=long, first-person=true, exclamation=true] **Human:** *I ordered the fried chicken sandwich with the french fries and it was fantastic, chicken was super juicy in a yummy brioche bun!*	**235k**	**41k**

Table 1.4: Sample MR and NL from popular NNLG datasets compared to YelpNLG.

1.2.3 Controllable Style in NNLG

Given our new, massive, stylistically varied YELPNLG corpus, we are now in a unique position to, for the first time, explore whether we can develop NNLG models that are able to produce NL that is more similar to natural human language style than the state-of-the-art.

Previous work in NNLG has attempted to *individually* control some aspects of style while focusing on single style attributes such as formality and verb tense,

MR 1	**(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.
STYLE	I had the gyro salad and the meat was so tender and juicy that it melted in your mouth.

MR 2	**(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.
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 1.5: Sample model outputs demonstrating joint control of multiple aspects of style.

sentiment, 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], and customer care dialogues [Herzig et al., 2017], but no previous work has attempted to model a more general spectrum of stylistic parameters in NNLG. Our YelpNLG corpus, which organically contains thousands of examples of creative language in the highly descriptive restaurant review domain allows us to model complex interacting stylistic constructions, in a way that previous work could not.

In Chapter 6, we present an ablation-style study aimed at testing how adding increasing style information into our training MRs as supervision to state-of-the-art NNLG models affects the amount of stylistic variation we are able to control in our outputs. We present 4 different experiments, each adding in additional information to our models, focused on *adjectives*, *sentiment*, and *style*, and conduct a rigorous set of quantitative and qualitative evaluations to explore how well we are able to control specific structural and stylistic phenomena we are interested in. Our experiments show, for the first time, that we are able to control multiple interacting aspects of style in our outputs; from lexical word choice (a completely novel contribution to NNLG), to sentiment, length, and the use of personal pronouns.

Table 1.5 shows examples of outputs from our BASE (no style information) and STYLE (full style information) models, showing how much more diverse the style outputs are, both satisfying semantic constraints in terms of value realization, and, for the first time, clearly hitting multiple style targets correctly. We are able to produce exciting stylistic choices from our models: for example, we see the addition of the phrase *"so tender and juicy that it melted in your mouth"* produced by the STYLE model in MR 2, as a way to simultaneously hit the sentiment and length targets. We also present an analysis characterizing how much variation we observe in each model output in Chapter 6.

Through these experiments on corpus creation and model design, we are able to show that we can create a corpus for NLG using exclusively off-the-shelf tools and resources, and that we can then use the richness of this dataset to systematically control different interacting aspects of style in our model outputs. Our most stylistically varied outputs are judged to be competitive for content preservation, fluency, and correct sentiment, and we are able to show that adding in this additional style information as "conditioning" to the model has the effect of helping to control semantic errors. Thus, we are able to achieve our aim of simultaneously hitting both semantic and stylistic targets.

1.3 Summary of Contributions

In summary, we approach this thesis by first systematically proving that stylistic control of NNLG is possible, using the PersonageNLG synthetic corpus that we create. We demonstrate that with some supervision, an NNLG model can learn to reproduce simple stylistic choices characteristic of the training data. Then, we design a method to take advantage of freely available, abundant, and rich human generated content, shaping it into a form we can use for training NNLGs. We present YelpNLG, the largest, most stylistically diverse corpus of MRs and matching NL for NNLG to date. Finally, we use our new corpus to produce novel, highly varied NL, jointly controlling multiple aspects of style, as judged by detailed quantitative and qualitative evaluations. Our contributions are:

1. Two large-scale corpora specifically designed for stylistic variation in NNLG:
   (a) PersonageNLG: A corpus of 88,000 MRs to reference texts based on the E2E Generation Challenge dataset and corresponding to stylistic properties of Big Five Personalities, including a variety of marked aggregation and pragmatic marker operations.
   (b) YelpNLG: A corpus of 300,000 MRs to reference texts created using freely available restaurant reviews from Yelp. MRs include semantic information as well as a novel characterization of descriptive lexical choice, sentiment, length, personal pronouns, and exclamations.
2. A methodology for creating corpora for NLG using freely available data and off-the-shelf tools, scalable to different domains.
3. A novel set of experiments on controlling multiple interacting aspects of style

with an NNLG while maintaining semantic fidelity, including personality, lexical choice (i.e. adjectives), sentiment, length, and personal pronouns. Results include a rigorous analysis of semantics and style using a broad array of evaluation metrics to supplement traditional lexical measures, including entropy, readability, sentence length, vocabulary and adjective counts, and contrast and aggregation analysis.

1.4 Thesis Outline

This thesis begins with a description of relevant previous work from the literature in Chapter 2, as related to NLG as a whole and the study of style in NLG with both statistical and neural models. Chapter 3 gives an overview of neural models for NNLG, based on the E2E Generation Challenge, as background for the methods and experiments we present in this thesis.

In Chapter 4, we present our first set of experiments for inducing structural and stylistic variation in NLG using PersonageNLG, a corpus of personality-based stylistic variation that we design. Chapter 5 describes our novel methodology for creating corpora for NLG using freely available data, and presents our YelpNLG corpus for stylistic variation in the restaurant review domain. Finally, Chapter 6 describes various methods for controlling NNLG systems, showing how we are able to jointly control semantics, style, and sentiment in our YelpNLG corpus. We conclude and describe applications and future directions in Chapter 7.

Chapter 2

Previous Research

2.1 Overview

This thesis aims to allow neural generators to produce controllable and varied output. To this end, we begin our exploration of previous work with an introduction to the study and goals of Natural Language Generation (NLG) systems, first examining the traditional pipeline for statistical NLG systems, then moving on to discuss state-of-the-art neural NLG architectures.

For both statistical and neural models, data collection for NLG is a longstanding problem, and is perhaps amplified by the need for massive datasets to train End-to-End (E2E) neural models which claim to learn directly from the data without the need for rules or intermediate representations. We thus enumerate various methods for data collection, and existing datasets used in the literature.

A primary goal of this thesis is to allow neural models to not only generate output with high semantic fidelity, but to allow for structural and stylistic control within the generation pipeline. To better understand how our work builds on existing work on introducing style into NLG models, we discuss previous work on statistical and neural style generation.

Finally, we discuss evaluation measures in NLG, which is another critical NLG concern due to the inherent difficulty of empirically evaluating utterances in a systematic and reusable way. We present a discussion of the history of evaluation measures used in NLG, comparing different methods frequently used in the literature.

2.2 Natural Language Generation Overview

NLG systems are primarily designed to convert some communicative goal into natural language that can be read or spoken. Methods for NLG most commonly begin with some form of structured plan, frequently referred to as a meaning representation, which describes the content to be expressed.

The concept of a Meaning Representation (MR) is not unique to the problem of language generation. Work on transfer-based machine translation refers to the Vauquois triangle, shown in Figure 2.1, which exposes the need to have an intermediate representation of "meaning" to generate a correct translation [Vauquois, 1968].[1] Similarly, work on summarization [Barzilay, 2003] and sentence simplification [Siddharthan, 2004] identify the need of an underlying representation for semantic control. There is also a long line of work on semantic parsing using Abstract Meaning Representations (AMR) targeting various different applications [Banarescu et al., 2013, Dorr et al., 1998, Flanigan et al., 2014]. Very recent work also proposes Dependency Minimal Recursion Semantics (DAMR) [Hajdik et al., 2019] for

[1] Image from Wikipedia, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=683855

capturing syntax information along with the semantic information in an AMR.

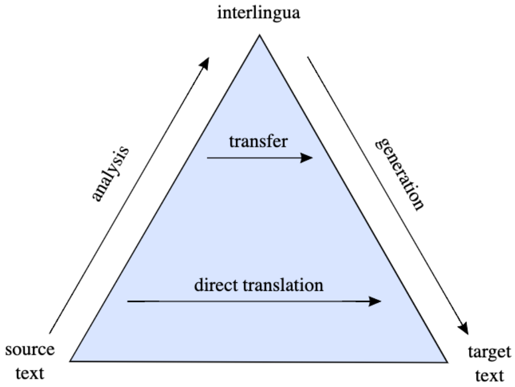


Figure 2.1: Vauquois triangle of intermediate representation in translation [Vauquois, 1968].

MRs are especially important in the context of NLG systems, which may be applied in a variety of different contexts, such as within a dialog system: in this case, it is critical for the system to have a coherent representation of goals, assertions, and dialog state, and consistency is essential. Thus, there is an implicit assumption that having an MR will lead to NLG systems that are more contextually aware, and will interface more consistently with components such as dialog managers.

Given some representation of meaning and content, the goal of the NLG system is then to map the content into a surface language form, which has been done using a range of different methods, including template-based generation, statistically trained NLG engines, and neural approaches [Bangalore and Rambow, 2000, Walker and Rambow, 2002], as we will discuss in this section.

The simplest NLG systems are template-based, which use rules, templates, and grammars to define how the output is shaped from content to Natural Language (NL) string. While these methods are simpler, faster to develop, and more easily customizable than other methods for NLG, they suffer from coverage and adaptation problems: it is difficult to scale a system that requires manual rules, and new domains frequently require new rules altogether [Bangalore and Rambow, 2000, Rudnicky et al., 1999]. There have been some efforts to generate more expansive rule-based modules that are more general-domain [Lavoie and Rambow, 1997, Elhadad and Robin, 1996], but very generic coverage and open-domain use is still a notable limitation, which is why trainable statistical and neural models are the current state-of-the-art for NLG.

In this section, we will discuss some of the most prevalent methods used to build NLG systems.

2.2.1 Statistical Models

Statistical models for NLG generally come in two types: modular pipelined architectures, and end-to-end architectures. Pipelined models consist of distinct modules to perform separate tasks, which we describe in more detail here, while end-to-end models generate from content to natural language essentially in a single step, without intermediate representations [Konstas and Lapata, 2013]. The advantage of a modular pipeline is that each stage has a well-defined role and can potentially be reused across systems, but end-to-end models may be less complex, reducing the need for detailed modeling and eliminating accumulated pipeline error [Dusek et al., 2019]. We describe each type here, detailing each individual task.

Pipelined Statistical Models

The most traditional statistical NLG models are built around a modular architecture, consisting of pipelined stages to convert an input communicative goal into a final surface string, including high-level content planning, individual sentence

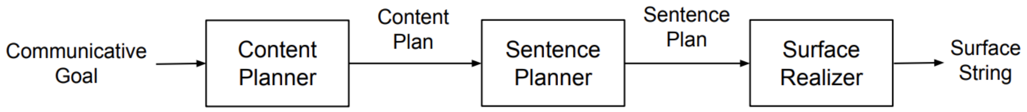


Figure 2.2: Traditional NLG architecture [Mairesse, 2008, Reiter and Dale, 2000].

planning, and surface realization, as shown in Figure 2.2 [Reiter and Dale, 2000, Rambow et al., 2001, Stent, 2002, Stent and Molina, 2009].

Table 2.1 outlines the tasks performed by each module. In general, the content planning phase is in charge of selecting the required content and performing basic structuring operations, dictating *"what to say"* for a given goal, and surface realization describes *"how to say"* the content, linearizing the content based on the output language, resulting in a natural language string. Sentence planning joins together both tasks, providing a more detailed semantic and syntactic representation for sentences in the output [Meteer, 1990, Dale et al., 1998, Dusek, 2017]. Since the sentence planner is traditionally in charge of the stylistic choices in the output [Stent et al., 2004], which is what we are most interested in this thesis, we defer a more detailed description of its role to Section 2.4, where we discuss work on stylistic variation in NLG.

One big advantage of a modular statistical pipeline as opposed to a rule-based one is the ability to develop individual modules that are adaptable and trainable. For example, there has been work on trainable content planners, which learn things like ordering constraints and global coherence from large data corpora [Marcu, 1997, Lapata, 2003, Barzilay and Lapata, 2005]. There is also a great deal of work on trainable sentence planners [Barzilay and Lapata, 2006, Stent and Molina,

Module	Task Description
Content Planner	Takes in the communicative goal and performs content selection (choosing propositional elements to express), and rhetorical structuring (defining which discourse relations will be used to express propositions, like "contrast" or "justify"). The output is a structured representation of what is to be expressed.
Sentence Planner	Converts the content plan into a syntactic representation that describes the structure of the sentence to be realized. This includes content ordering, selecting syntactic templates, aggregation (what words will be used to express the rhetorical propositions supplied by the content planner), and word and phrase level lexical choices.
Surface Realizer	Transforms the syntactic representation output by the sentence planner into a final surface string of text by applying language rules.

Table 2.1: Tasks performed by each module in traditional statistical generation.

2009, Walker et al., 2007, Sauper and Barzilay, 2009, H. Cheng and Mellish, 2001], including HALOGEN [Langkilde and Knight, 1998], SPOT [Walker et al., 2001], and SPARKY [Stent et al., 2004] which frequently employ an overgenerate-and-rank approach where multiple sentence plan candidates are generated and then the "best" are selected by some ranking criteria. Other methods reduce the cost of overgenerate-and-rank planners by implementing parameter estimation methods or doing factor analysis [Mairesse, 2008, Paiva and Evans, 2004]. Statistical and hybrid methods using overgenerate-and-rank have also been developed for surface realization [Langkilde and Knight, 1998, Belz and Reiter, 2006, Bangalore and Rambow, 2000, Oh and Rudnicky, 2002].

End-to-End Statistical Models

While trainable methods for Statistical Natural Language Generation (SNLG) provide a partial solution to the scalability problem over rule-based methods, they are still constrained by the need for predefined syntax and handcrafted definitions of the decision space for statistical optimization in generation [Wen et al., 2015].

Thus, the most recent movement in NLG has been towards E2E models that learn directly from data, abstracting away from a pipelined architecture, and performing the sentence planning and surface realization stages jointly in a single step.

There has been some work on data-driven statistical models for E2E NLG, primarily based on n-gram language models, which use a Markov model to predict the probability distribution of the next token in a sequence based on the previous $n-1$ tokens. These methods typically use a combination of handcrafted and statistical units to generate phrases by performing a word-by-word beam search [Oh and Rudnicky, 2002, Ratnaparkhi, 2000, Ratnaparkhi, 2002, Wong and Mooney, 2007]. Other work focuses on training models to choose the best template to realize [Angeli et al., 2010], with updates using SVM re-ranking [Kondadadi et al., 2013] for improved selection. More recent work uses both a language model and semantically-aligned corpus to do phrase-based NLG [Mairesse and Young, 2014]. Despite these improvements to fully rule-based models, reliance primarily on template matching means generated output still does not generalize well [Wen et al., 2015], and the need for semantically-aligned data for generation [Mairesse et al., 2010, Konstas and Lapata, 2013] puts an added burden on already-arduous corpus creation methods based on crowd-sourcing (which we describe in more detail in Section 2.3).

2.2.2 Neural Models

With the success of neural models on tasks such as machine translation [Cho et al., 2014, Sutskever et al., 2014, Bahdanau et al., 2014], speech recognition [Mikolov et al., 2013], and image captioning [Vinyals and Le, 2015], there has been a huge movement towards using neural models for E2E NLG. This paradigm currently serves as the state-of-the-art for data-driven NLG [Mei et al., 2016, Wen et al., 2015, Dusek and Jurcícek, 2016, Lampouras and Vlachos, 2016, Nayak et al., 2017, Juraska et al., 2018, Oraby et al., 2018b, Reed et al., 2018].

The neural approach to NLG promises to simplify the process of producing high quality natural language in any domain by relying on the neural architecture to automatically learn how to map required input content to output natural language realizations, only requiring parallel corpora of MR to NL as input (as we showed in Table 1.1), and leaving all language modeling for the Neural Natural Language Generation (NNLG) to learn implicitly from the data. This eliminates the need for any handcrafting or costly semantic alignment as was required for SNLG methods.

Neural models for various language tasks generally include a conditioned Recurrent Neural Network (RNN) language model [Mikolov et al., 2011] to replace the n-gram based language model from previous work. The RNN language model uses an RNN [Rumelhart et al., 1986, Goodfellow et al., 2016], a class of neural network designed to process sequential data, to predict next-token probabilities based on *all previous tokens*, which allows for long-distance dependencies. The RNN language model is thus capable of modeling more information than the n-gram based models previously used in SNLG systems, which only models short-term dependencies for the $n-1$ preceding tokens based on corpus statistics [Dusek, 2017].

In the case of NLG and other related sequence generation tasks, the standard approach is to use a Sequence-to-Sequence (seq2seq) model, first popularized by Cho et al. [Cho et al., 2014] and Sutskever et al. [Sutskever et al., 2014]. The general idea is that an input sequence of tokens is input token-by-token by an *encoder* RNN, which encodes it into a hidden state that is consumed by a *decoder*

RNN to produce an output sequence [Dusek, 2017]. The most commonly used RNN cell is the Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997], since it does the best job at modeling long-term dependencies. We describe the architecture of state-of-the-art NNLG systems in detail in Chapter 3.

Given this overview of the history of NLG models from template-based, to statistical, to neural models, we move on here to describe the most popular datasets used for training recent NLG systems.

2.3 Datasets for Natural Language Generation

Early works on trainable NLG systems used small datasets collected by eliciting NL responses/paraphrases from users in some situational context. In general, crowdsourcing has frequently been used in evaluation of NLG systems, and there has been a growing trend of using various crowdsourcing-based methods to collect high-quality datasets for training NLG systems. Table 2.2 enumerates some of the most popular datasets for NLG, which we describe in more detail in this section. The datasets in Rows 1-2 were designed for SNLG systems, while the rest were designed for NNLG systems. The datasets in Table 2.2 are sorted by size, but we point to the fact that the datasets have grown significantly in size over the years, as NNLG systems become state-of-the-art.

The BAGEL dataset of restaurant descriptions from Mairesse et al. [Mairesse et al., 2010] is an early example of using crowdsourcing to create a dataset for NLG (Row 1, Table 2.2).[2] To elicit high-quality data, the task was situationally

[2] BAGEL dataset: `http://farm2.user.srcf.net/research/bagel/`

#	Dataset	Domain	Collection Method	Size
1	BAGEL [Mairesse et al., 2010]	Restaurants	Crowdsourcing	400
2	RoboCup [Chen and Mooney, 2008]	Sportscasting	Crowdsourcing	2k
3	SF Hotels [Wen et al., 2015]	Hotels	Crowdsourcing	5k
4	SF Restaurants [Wen et al., 2015]	Restaurants	Crowdsourcing	5k
5	TV [Wen et al., 2016]	Tech Reviews	Crowdsourcing	7k
6	Laptop [Wen et al., 2016]	Tech Reviews	Crowdsourcing	13k
7	WebNLG [Gardent et al., 2017a]	Wikipedia	Crowdsourcing	25k
8	E2E NLG [Novikova et al., 2017b]	Restaurants	Crowdsourcing	50k
9	**PersonageNLG** [Oraby et al., 2018b]	**Restaurants**	**Synthetic Augment.**	**88k**
10	**YelpNLG** [Oraby et al., 2017, Oraby et al., 2019]	**Restaurants**	**Automatic Extraction**	**300k**
11	WikiBio [Lebret et al., 2016]	Biographies	Wiki InfoBox + Para. 1	700k

Table 2.2: Recent datasets for NLG (Rows 1-2 were designed for SNLG systems, the rest for NNLG systems). Datasets resulting from this thesis are shown in bold.

framed: users are asked to write utterances that a "tourist information system" might produce in a dialogic context. The crowdsourcing task is divided into two tasks: response writing and alignment. First, crowd workers are shown an MR in the restaurant domain, including a communicative goal (to *inform*) and attribute-value pairs (9 different attribute types, e.g. *food*, and *area*), and asked to produce an NL realizing the required content.[3] As a second step, crowd workers are also asked to do an alignment task, where they must indicate *where* each MR attribute is realized in a given response.[4]

Some guidance was given to the writers to try to enforce some variation: for example, they were encouraged to vary lexical choice for some attribute types (for example, "city centre" could be realized in any way they wished to express the

[3] Task 1, writing: http://farm2.user.srcf.net/research/bagel/Phase1Example1.html

[4] Task 2, alignment: http://farm2.user.srcf.net/research/bagel/Phase2Example1.html

intended meaning), they were reminded to use any punctuation they might prefer, and they were not asked to realize all content items in any particular order. To ensure quality, the utterances produced were manually checked for correctness and alignment. Also, as is common for NLG datasets, the utterances were delexicalized (i.e. instances of "non-enumerable values" such as restaurant names were replaced with a placeholder, e.g. *"X"*) to reduce some unnecessary data sparsity and allow for more generic language templates as input to the generator [Dušek and Jurcıcek, 2016, Dusek, 2017]. The resulting dataset includes 404 written instances, along with their respective manual alignments.

RoboCup [Chen and Mooney, 2008] is another example of a popular dataset for NLG, this time based on human commentaries collected for robot soccer games in the Robocup[5] simulation league (Row 2, Table 2.2).[6] Users were asked to write their commentaries during the games, and each of these NL comments were marked with a timestamp. Game events (e.g. *kicking* or *passing*) were extracted from game logs using a rule-based algorithm, and MRs were constructed as a formal semantic language in predicate logic form (e.g. *pass(pink1, pink4)* to define a passing event), and were then paired with the human commentary using the timestamp information (with a subset manually matched to construct a gold-standard set). The final RoboCup dataset consists of around 2k instances.

While BAGEL and RoboCup are examples of good quality, carefully constructed datasets for NLG, the recent shift to using neural methods for NLG introduces new requirements for a much larger number of training instances than these datasets provide, to avoid problems such as overfitting. Rows 3-11 in Table

[5] https://www.robocup.org/

[6] RoboCup dataset: http://www.cs.utexas.edu/users/ml/clamp/sportscasting/

2.2 show examples of datasets designed for use in an NNLG setting that are notably larger in size than previous work. Although NNLG systems require more data than SNLG systems did, no alignment between the instances in the MR and the NL reference is necessary. The neural model jointly learns to align and generate [Bahdanau et al., 2014], as we describe in Chapter 3, and all that is required are large parallel corpora of MR to NL.

Wen et al. present the SF HOTELS and SF RESTAURANTS datasets [Wen et al., 2015], each containing around 5k instances of MR to NL pairs for training early NNLG systems (Rows 3-4, Table 2.2). The datasets are collected through crowdsourcing in the context of a dialog system about venues in San Francisco, where crowd workers are shown a dialog turn-by-turn and are prompted with an MR, which they are asked to write a NL realization for. The MRs include one of 8 different dialog acts (i.e. communicative goals, like *inform* or *request*), and any of 12 different attributes (9 of which are shared between the hotel and restaurant domains). After delexicalization, the number of distinct MRs is actually only 248 for restaurants and 164 for hotels, with an average of 2.25 and 1.95 attribute-value pairs per MRs for restaurants and hotels, respectively [Wen et al., 2015].[7]

In 2016, Wen et al. also released two new and larger datasets for NNLG: the TV and LAPTOP product review datasets [Wen et al., 2016], which are around 7k and 13k instances in size, respectively (Rows 5-6, Table 2.2). These datasets contain much more variation than SF HOTELS and RESTAURANTS, and are elicited through crowdsourcing as before. There are 14 dialog act types, with 15 different

[7] Wen et al. do not specify exactly which attribute-value pairs they delexicalize, but summarize that they delexicalize all values that are not binary (i.e. not the attribute "kid friendly" since the value could be "yes—no"), or attributes that an take on a value of "don't care" [Wen et al., 2015, Dusek, 2017].

attributes for TV and 19 for LAPTOP.

Other large datasets for NLG explore the use of some structured data available from sources such as DBPedia and Wikipedia to help create their dataset. The WEBNLG dataset, from Gardent et al. [Gardent et al., 2017a, Perez-Beltrachini et al., 2016], was designed for the WEBNLG challenge (Row 7, Table 2.2).[8] The dataset and challenge focus on generating text from RDF (Resource Description Framework) triples defining semantic relationships between entities in the Wikipedia domain, for example, *(John E. Blaha, occupation, Fighter Pilot)*. The dataset consists of 25k pairs of DBPedia triples and a corresponding crowdsourced NL of each triple.

The WIKIBIO dataset from Lebret et al. [Lebret et al., 2016] uses structured data from Wikipedia in a different way: the dataset consists of 700k Wikipedia infoboxes from biographical articles, paired with the first paragraph of the article (Row 11, Table 2.2).[9] We note that while WIKIBIO is the largest currently available dataset for NLG, the data is inherently noisy, since there is no guarantee of coverage or matching between the infobox and first-paragraph NL.

More recent work from Novikova et al. [Novikova et al., 2016, Novikova et al., 2017b] has focused on other creative ways for crowdsourcing large and diverse datasets for NNLG (Row 8, Table 2.2). They focus on the restaurant domain, as with the BAGEL [Mairesse et al., 2010] and SF RESTAURANTS [Wen et al., 2015] datasets, but they use *pictorial* representations of restaurant MRs to elicit user descriptions, rather than textual MRs. They validate their method by first collecting and analyzing 1.4k varied restaurant descriptions [Novikova et al., 2016],

[8] http://webnlg.loria.fr/pages/challenge.html

[9] Dataset: https://github.com/harvardnlp/neural-template-gen

then expand the dataset to 50k instances, released in the widely popular E2E NLG Dataset Challenge [Novikova et al., 2017b].[10] The idea behind the use of pictorial MR representations instead of traditional text-based ones is to elicit more diverse language: users have been found to be "primed" and limited by the word choice and ordering shown in text-based MRs [Wang et al., 2012, Novikova et al., 2016]. We describe the E2E NLG dataset, collection method, baseline models, and challenge findings in more detail in Chapter 3, as it relates to our own work on NNLG.

Although previous work has almost *exclusively* used crowdsourcing to collect datasets for NLG, this method is inherently limited. First, crowdsourcing very large datasets (as required for NNLGs) requires time and money: in fact, it is difficult to adjust pay for crowd workers to maximize quality, and some studies have even shown an *inverse* relationship between payment amounts and work quality [Mason and Watts, 2009, Callison-Burch and Dredze, 2010]. Secondly, quality control is a difficult endeavor; for example, in creating the E2E dataset, Novikova et al. perform a two stage quality control method, including "automatic pre-validation" and "human evaluation" [Novikova et al., 2016]. Additionally, even with creative methods for crowdsourcing such as the use of pictorial MRs, crowd workers are inherently influenced by the presentation method [Wang et al., 2012], and it is difficult to incentivize them to produce sufficiently varied and diverse realizations when prompted in a crowdsourcing setting. Finally, it has been shown that the semantic complexity that crowd workers can handle is limited [Mairesse et al., 2010].

In our own work, we focus on novel, non-crowdsourcing methods to produce large and diverse datasets for NNLG. Our first dataset, PersonageNLG (Row 9, Table 2.2), is an extension of the E2E dataset, where we synthetically generate

[10] Challenge website: `http://www.macs.hw.ac.uk/InteractionLab/E2E/`

additional NL realizations for training MRs by using the PERSONAGE statistical generator [Mairesse and Walker, 2007], which takes in input MRs and outputs different ways of expressing the content in Big-Five personalities. PERSONAGENLG contains 88k instances, as compared to the 50k instances in E2E NLG, and is widely more varied, with numerous examples of interesting personality-specific aggregation operations and pragmatic marker usage. We discuss our PERSONAGENLG dataset in detail in Chapter 4.

While PERSONAGENLG allows us to produce novel examples of style in NLG, it is still limited by the use of a synthetic data generator that produces a finite set of possible stylistic choices. Since of course the output we expect to generate with an NNLG is a result of what the models see in training, what we really need are datasets that provide models with enough examples of complex language phenomena to allow for more diverse generation, which is a long-standing interest of the NNLG community [Walker et al., 2004, Stent et al., 2004, Demberg and Moore, 2006, Rieser and Lemon, 2010].

Motivated by a desire to produce a dataset that is as varied as real human language, we produce the YELPNLG dataset, which contains 300k paired MR to NL rich examples from the restaurant review domain (Row 10, Table 2.2). We create YELPNLG without *any* crowdsourcing by working in reverse: we begin with naturally occurring user reviews, do some preprocessing to select sentences to use as our NL examples, and then automatically generate MRs using off-the-shelf knowledge bases of different restaurant attributes of interest, such as *food*, *service*, and *staff*. We describe our method for creating YELPNLG in Chapter 5, where we show how much richer both the naturally-occurring NL and our automatically-generated MRs are, as compared to any existing work on NLG. We also show how our corpus creation method can be extended to other domains, providing a new way of generating good quality corpora for NLG from freely-available data without crowd sourcing.

In the next section, we discuss existing work over the years on generating stylistically varied realizations, using the datasets we have described here, also explaining how our own work helps tackle the style gap in NNLG.

2.4 Style in Natural Language Generation

In this thesis, we are most concerned with control of structural and stylistic variations in output text, which are traditionally controlled by the sentence planner [Stent et al., 2004]. We begin this section with a more detailed review of the tasks of the sentence planner (which were summarized in Section 2.2 in our overview of NLG), then move on to describe previous work on generating style in NLG.

The tasks of the sentence planner are presented in detail in Table 2.3, clarifying how the choices made at this stage in the pipeline affect the structure and style of the output. For example, sentence scoping affects the complexity of the sentences that compose an output, allowing the generator to produce simpler sentences when desired that might be easier for particular users to understand. Aggregation, which dictates how content is allocated within a single sentence, can reduce redundancy, composing multiple content items compactly into single sentences [Cahill et al., 2001, Shaw, 1998]. Discourse structuring is often critical in persuasive settings such as recommending restaurants, hotels or travel options [Scott and de Souza, 1990, Moore and Paris, 1993], in order to express discourse relations that

Task	Description
Content Ordering	Deciding in what order content should be expressed, (*e.g. "excellent decor and superb food", or "superb food and excellent decor"*).
Sentence Scoping	Deciding how to allocate the content to be expressed across different sentences, (*e.g. expressing all content in a single sentence or distributing it across multiple sentences*).
Aggregation	Implementing strategies for removing redundancy and constructing compact sentences, (*e.g. "excellent service and staff" instead of "excellent service and excellent staff"*).
Discourse Structuring	Deciding how to express discourse relations that hold between content items, such as causality, contrast, or justification, (*e.g. contrast, "the food was great but the atmosphere was awful", or justification "it is the best restaurant because it has amazing food"*).
Pragmatic Variation	Making decisions about variations in syntactic form, use of pragmatic markers and discourse cues, (*e.g. using hedges, like "well", or "i see"*).
Lexical Choice	Making decisions about which words or phrases to use to express a particular concept, (*e.g. expressing that the food is good in different ways: "the food was phenomenal", or "the food was the best I've ever had"*).

Table 2.3: Tasks performed by the sentence planner in traditional statistical generation.

hold between content items [Stent et al., 2002]. Pragmatic variation, such as the use of different pragmatic markers or discourse cues, and lexical choice, or which words to use to express concepts, also have a clear effect on the perceived style of the output.

The restaurant domain is an ideal testbed for these sentence planning operations for generation models, because sentences naturally range from extremely simple to more complex forms that clearly exhibit the discourse relations that the sentence planner is in charge of producing [Stent et al., 2004]. For this reason, there has been a tremendous amount of previous work on natural language generation of recommendations and descriptions for restaurants, both in terms of datasets (see Table 2.2), and for SNLG and NNLG systems [Stent et al., 2004, Devillers et al., 2004, Gašic et al., 2008, Mairesse et al., 2010, Higashinaka et al., 2007b, Howcroft et al., 2013, Wen et al., 2015, Mei et al., 2016, Novikova et al., 2017b, Dusek and Jurcícek, 2016, Lampouras and Vlachos, 2016, Oraby et al., 2017, Oraby et al., 2019, Oraby et al., 2018b, Reed et al., 2018, Juraska et al., 2018].

A few pieces of work on SNLG attempt to introduce stylistic variation into generated utterances [Higashinaka et al., 2007b, Mairesse and Walker, 2010, Dethlefs et al., 2014, Paiva and Evans, 2004, Isard et al., 2006, Inkpen and Hirst, 2004]. Early work on stylistic variation in NLG was conducted in the context of a modular SNLG system, where the sentence planner is a separate module that is governed by parameterized rules. Although it is theoretically possible for these types of SNLG system to control the types of stylistic operations produced within generated output, they are severely limited by a need for primarily manually-defined rules.

Some of this previous work has used a hybrid approach involving a linguistic core with underspecified parameters, which were then learned from training data [Mairesse and Walker, 2010, Dethlefs et al., 2014]. An alternative method harvested NLG templates from user-generated restaurant reviews, and then incorporated them into the linguistic core of the SNLG system [Higashinaka et al., 2007b].

Table 2.4 illustrates some sample restaurant domain utterances produced by recent statistical/neural natural language generators [Higashinaka et al., 2007a, Mairesse and Walker, 2007, Wen et al., 2015, Novikova et al., 2016, Dusek and Jurcícek, 2016], to exemplify the problem of style in NLG. In terms of SNLG, Rows 1-2 in Table 2.4 show examples where the required content items are realized with a few stylistic additions, such as the use of pragmatic markers and hedges [Mairesse and Walker, 2007, Higashinaka et al., 2007a]. Specifically, we see examples of justification in Row 1, with *"since the service is fast..."*, as well as varied lexical choice, with *"the best overall quality"* to express the *foodquality[superb]* content item. In Row 2, we see an example of aggregation with *"and"*, and pragmatic marker *"actually"*, as well as contrast, with *"the food is good, even if the price is ..."*.

Examples of previous work in NNLG in the restaurant domain, shown in Rows 3-4, are much less stylistically diverse, focusing primarily on semantic fidelity, without a clear mechanism on how to introduce real stylistic variation into the generation pipeline [Dusek et al., 2019]. However, we do still see simple examples of aggregation, using the *"and"* operator, used to aggregate across different values of the same attribute (*"food"*) in Row 2, and different attributes (*"area"* and *"kids-allowed"*) to express them in the same sentence in Row 4.

Rows 5-6 show examples from our own work on introducing stylistic variation into the generation pipeline of NNLG systems. In Row 5, we show an example generated with an NNLG system trained on our PersonageNLG corpus from Chapter 4, which shows that we are able to produce outputs that exhibit the types of aggregation and pragmatic marker choices (e.g. *"well", "i see"*, and tag question *"okay?"*) as in the SNLG outputs from Rows 1-2. Row 6 shows an example output from our NNLG system trained on our YelpNLG corpus on restaurant reviews, which we describe in Chapter 5, including strong and interesting lexical choices, such as *"the best I have ever had"* to express positive sentiment. With the richest MRs seen to date in work on NLG, we are able to simultaneously control semantic and style choices: producing outputs that are stylistically rich, and adhere to stylistic constraints from the MR, as we show in Chapter 6.

In contrast to the notion of parameterized style choices in the world of

#	System	MR	NLG Output
1	SNLG Unsupervised Method for Lexicon Learning [Higashinaka et al., 2007a]	name[Babbo] foodquality[superb] service[excellent] decor[superb]	**Since** *the service is fast and friendly, the food is really good and Babbo has excellent decor, it has* **the best overall quality** *among the selected restaurants.*
2	SNLG Personage [Mairesse and Walker, 2007]	name[Le Marais] cuisine[French] foodquality[good] service[good] decor[decent] price[$44]	*I am sure you would like Le Marais, you know. The atmosphere is acceptable, the servers are nice* **and** *it's a french, kosher and steak house place.* **Actually**, *the food is good,* **even if** *its price is 44 dollars.*
3	NNLG Seq2Seq Generation [Dusek and Jurcícek, 2016]	name[X] type[placetoeat] near[Y] food[Continental] food[French]	*X is a restaurant providing* **french and continental food**, *near Y.*
4	NNLG Seq2Seq Generation [Wen et al., 2015]	name[Red door cafe] goodfor[breakfast] area[cathedral hill] kidsallowed[no]	*Red door cafe is a good restaurant for breakfast in the area of cathedral hill* **and** *does not allow children.*
5	NNLG Seq2seq Generation with Personage data [Oraby et al., 2018b, Reed et al., 2018]	name[The Eagle] eattype[restaurant] food[Italian] familyFriendly[yes] pricerange[moderate] customerRating[high] area[city centre] near[Burger King] personality[agreeable]	**Let's see** *what we can find on The Eagle.* **Well, i see** *it is a restaurant, also it is an italian restaurant, in city centre moderately priced near Burger King and family friendly,* **okay?**
6	NNLG Seq2seq Generation with Yelp data ([Oraby et al., 2017, Oraby et al., 2019])	(food, meat, fresh, mention=1), (food, vegetables, no-adj, mention=1), (food, bread, no-adj, mention=1), (sentiment=positive, length=long, first-person=true, exclamation=false)	*The meat was fresh, the vegetables were cooked perfectly, and the bread was* **the best I have ever had**.

Table 2.4: Example system outputs in the restaurant domain, with some stylistic variation.

SNLG, most previous work on style generation in NNLG has been carried out in the framework of "style transfer", where text is generated from other text, as in translation or paraphrase (i.e. text-to-text vs. data-to-text as in our own work).

This work attempts to generate diverse outputs by learning style choices directly from the data, and does not have the benefit of MRs for defining content. The lack of an MR for content representation has made this type of work hard to evaluate, since it is unclear how to evaluate semantic fidelity in this setting.

For example, Shen et al. [Shen et al., 2017] focuses on style transfer on Yelp reviews, aiming to convert the sentiment of a review sentence to the opposite polarity by trying to learn a "style transfer function" from non-parallel data. The main challenge here is to disentangle content from style, which is a problem introduced here due to the absence of any semantic grounding (i.e. MR) in this case. Other experiments attempt to control the sentiment and verb tense of generated movie review sentences [Hu et al., 2017] using a combination of generative variational auto-encoders and attribute discriminators to derive latent representations of semantics (again, directly from text, without an MR for content definition). We focus our own work on using a semantically-grounded MR to guide generation, as would be the natural setup in a dialog system required to express known content to a user.

Fu et al. focus on the content preservation and style transfer of news headlines and product review sentences, again using non-parallel data [Fu et al., 2018] to learn separate representations of style and content using adversarial networks. They find an inverse correlation between transfer strength and content preservation: indicating that within a model, if the goal is to preserve more style, some content will be lost. In our own work, we focus on avoiding this content-style trade-off with models that are able to simultaneously preserve semantics from a known input MR, but also generate interesting stylistic variations along interacting dimensions.

Other work has focused on style generation of different applications. For abstractive summarization, Fan et al. attempt to generate summaries with controllable length and topic, using a corpus of news articles-to-summaries [Fan et al., 2017]. In the customer care dialog systems domain, Herzig et al. train on customer service dialogs to adapt system responses to different Big-Five personality types (which we also use in our own work on generation with PersonageNLG corpus in Section 4.2) [Herzig et al., 2017].

More similar to our own goals, Ficler et al. attempt to control multiple automatically extracted style attributes along with sentiment and sentence theme for movie reviews [Ficler and Goldberg, 2017]. Similar to our own work on automatically generating corpora for training NNLGs, they use meta-data to annotate some attributes of interest in reviews, then experiment with conditioned and unconditioned language models.

While this work does attempt to control some aspects of semantics and style as we do, they are focused on a limited array of well-defined style attributes, while we are more interested in modeling a broader range of jointly interacting parameters of style, and on explicitly affecting features previously unexplored in NNLG, such as adjectival lexical choice. Also, without common semantic grounding in the training data, i.e. movies from a specific genre, it is difficult to control truth grounding in resultant realizations; we control this by training data that is highly semantically related in our own work (Section 5.3.2). Most critically, the authors cite the need for finer-grained content control in their models, which we focus on as a primary requirement in our own work, measured through rigorous quantitative evaluation (Sections 4.4.1 and 6.4.1).

Other work has also focused on different style dimensions. Rao and Tetreault generate variations along the formality dimension, introducing a large parallel corpus for formality, including a custom trained model for attempting to measure semantic fidelity [Rao and Tetreault, 2018]. Other work has focused on politeness [Aubakirova and Bansal, 2016, Sennrich et al., 2016]. However, to our knowledge, no previous work evaluates simultaneous achievement of multiple interacting style targets as we do (Sections 6.4).

As more work looks to produce more stylistically diverse outputs using both SNLG and NNLG systems, we are faced with another long-standing problem: the difficulty of evaluating the outputs in any meaningful way. It is not clear how to measure content preservation especially with lexically-varied outputs, and without clear style constraints, it is also unclear how to measure whether the outputs realize a particular style. In the next section, we describe current and historical methods for evaluating NLG model outputs.

2.5 Evaluation Metrics

There has been a great deal of research on designing, standardizing, and comparing good metrics for evaluating the output of NLG systems [Jones and Galliers, 1996, Mellish and Dale, 1998, Bangalore et al., 2000, Belz and Reiter, 2006, Hastie and Belz, 2014, Gkatzia and Mahamood, 2015, Novikova et al., 2017a, Gatt and Krahmer, 2018]. Still, there has yet to be a clear consensus on a standard way of evaluating NLG output that is robust across many systems and domains.

Belz and Reiter [Belz and Reiter, 2006] characterize evaluation measures used in NLG as *"intrinsic"* [Jones and Galliers, 1996, Bangalore et al., 2000], where system outputs are evaluated by comparing them to human-written ones, and *"ex-*

trinsic", or trying to measure the impact of system outputs on different tasks or within systems. They describe that both general types of traditional evaluations in NLG have been performed using human subjects, and these have more frequently been intrinsic, posed as questions that require human raters to compare system and human-written outputs. Their study mimics the findings of previous ones, such as that of Reiter et al. [Reiter et al., 2005], finding that correlation is low between experts asked to evaluate systems intrinsically, suggesting that extrinsic evaluation to gauge the effectiveness of system output in helping users perform tasks is also important.

More recently, Hastie and Belz [Hastie and Belz, 2014] further expand on the intrinsic/extrinsic taxonomy of NLG evaluation. They break down intrinsic evaluation into *"output quality measures"*, or evaluating the similarity between system output and some reference output (either automatically or through human ratings), and *"user-like measures"*, where users are asked to rate system outputs (usually on a Likert scale) based on some criteria. Similarly, they divide extrinsic evaluation into *"user task success metrics"*, where users are asked about the usefulness of system output for particular tasks (e.g. comprehension or decision making), and *"system purpose success metrics"*, where the system is evaluated based on whether the outputs fulfill the system's intended purpose [Gkatzia and Mahamood, 2015].

Gkatzia and Mahamood [Gkatzia and Mahamood, 2015] use Hastie and Belz's refined taxonomy in their study of evaluation trends in NLG from 2005-2014 based on a corpus of around 80 published papers in natural language processing conferences and journals. They estimate that the use of qualitative user-like measures are the most widely used evaluation metrics, due to how straightforward they are to collect, but they find that intrinsic user-like metric success does not necessarily correlate with extrinsic measures (as was found in previous work [Reiter et al., 2005]). Intrinsic measures such as output quality also suffer from the limited availability of reliable parallel corpora for NLG. Gkatzia and Mahamood point to advantage of repeatability of intrinsic output quality metrics [Belz and Reiter, 2006], and find that output quality metrics are frequently used in combination with other metrics (around 55% of the time in their corpus) to mitigate the effects of potentially erroneous "expert" references [Gkatzia and Mahamood, 2015].

The recent shift into training E2E neural generation systems from parallel corpora of meaning representations to reference texts has lead to a higher reliance on intrinsic evaluation in the form of quantitative automatic measures and qualitative user like measures. In this section, we cover some of the most commonly used measures for evaluation currently used in NLG systems.

2.5.1 Automatic Metrics

According to Gkatzia and Mahamood's study [Gkatzia and Mahamood, 2015], automatic metrics have been used in up to 60% of NLG research papers sampled between the years of 2012-2015 [Novikova et al., 2017a]. Given the increasing popularity of neural NLG systems, automatic evaluations are becoming even more mainstream, due to the need to quickly and cheaply benchmark systems, and for parameter tuning [Novikova et al., 2017a].

Word-based Metrics

The first type of automatic NLG metrics are n-gram based metrics borrowed from the machine translation, summarization, and image captioning communities. These metrics require comparison of a system output to one or more human references that represent a "ground truth", and a higher similarity score indicates "better" system outputs. Some commonly used automated metrics are listed below:[11]

- **BLEU:** [Papineni et al., 2002] A precision-based measure of n-gram overlap that computes the percentage of n-grams in the candidate output that appear in the ground truth references, with a length penalty. BLEU aims to answer the question: *"Of all of the n-grams in the candidate output, what percentage of them occur in the ground truth references?"*
- **ROUGE:** [Lin, 2004] A recall-based measure of n-gram overlap computed as the number of n-grams that occur in both the candidate and references, over the total number of possible n-grams in the references (i.e. based on longest common subsequences). ROUGE answers the question: *"Of all of the possible n-grams in the ground truth references, what percentage of them occur in the candidate output?"*
- **NIST:** [Doddington, 2002] An update to BLEU, using weighted n-gram precision (i.e. arithmetic average) in order to give a higher weight to less-frequent (more informative) n-grams.
- **CIDEr:** [Vedantam et al., 2014] TF-IDF based measure for weighted n-gram cosine similarity, derived from the image captioning community.
- **METEOR:** [Lavie and Agarwal, 2007] A precision and recall based measure of unigram overlap that allows for close-but-not-exact matching using word stemming and synonyms from WordNet by aligning the texts with human

[11] We summarize the metrics here, but give a more detailed description of them and how they are computed in our overview of neural NLG in Chapter 3.

references.

In order for these automatic word-based metrics to be useful for system benchmarking, their ratings should be highly correlated with human ratings of the same system outputs. However, this is frequently not the case [Stent et al., 2005, Belz and Reiter, 2006, Reiter and Belz, 2009], due to their inability to capture structural differences and slight semantic variations in compared texts.

In their detailed study on the need for new metrics for evaluation in NLG [Novikova et al., 2017b], Novikova et al. compare the performance of automatic metrics, similarity-based metrics such as Semantic Textual Similarity (STS) [Han et al., 2013], which has also been used to compare the "meaning" of texts, and human judgments, all scaled to values from 1 (lowest similarity) to 6 (highest similarity). Table 2.5 shows two sample MRs, each with a sample system outputs from an NNLG systems, and a human-written reference for two NNLG systems, TGen [Dušek and Jurcicek, 2015] and RNNLG [Wen et al., 2015]. In Row 1, we see that the system output is repetitive and does not include all the information in the MR, and is thus rated low by humans; however, it is given a high semantic similarity score, and 2.4/6 for the word-based metrics. In comparison, Row 2 shows a coherent, full output that is rated very well by semantic similarity measures and human judgments (both above 4/6), but poorly by the word-based metrics (1.85).

In their study, Novikova et al. conclude that while word-based metrics frequently agree with humans on "bad quality output", they do poorly on distinguishing between "good and medium quality" output [Novikova et al., 2017b]. The limitations of these metrics are also inflated in applications (such as ours) which aim to produce stylistic variation in generated output while preserving semantics.

#	MR	System Output	Human Reference	WBM	SEM	HUM
1 TGen	inform(name=X, area=riverside, eattype=restaurant, food=fastfood, pricerange=cheap)	x is a restaurant on the riverside called located at the riverside and at is	x is a cheap fastfood restaurant located near the riverside	2.4	4	1
2 RNNLG	inform-nomatch(kids-allowed=yes, food=moroccan)	i am sorry, i did not find any restaurants that allow kids and serve moroccan	sorry, there are no restaurants allowing kids and serving moroccan food	1.85	4	5

Table 2.5: Comparison between similarity scores given by word-based metrics (WBM), semantic similarity (SEM), and human judgments (HUM) for two MR to output pairs, normalized on a 1-6 scale [Novikova et al., 2017b].

Grammar-based metrics

Novikova et al. [Novikova et al., 2017a] proposed the use of grammar-based metrics for NLG evaluation, pointing to the fact that they do not require comparison with gold-standard references. Instead, these measures attempt to quantify more structural/stylistic properties of a candidate output. Two such measures are described below:

- **Readability:** A quantitative estimate of how difficult it is to read a given text. There have been several measures of readability in the literature; Novikova et al. [Novikova et al., 2017a] focus on Flesch Reading Ease [Flesch, 1997], which computes ratios between the number of characters and words in a sentence, and the number of syllables per word (which can each be used independently as properties of candidate texts).
- **Grammaticality:** Measures such as misspellings, parsing scores (Stanford parser [Chen and Manning, 2014]) are used to evaluate how "grammatical" a

text is. Other work aims to judge grammaticality automatically with a statistical model that uses linguistic features [Heilman et al., 2014]. Grammatically is important in the context of E2E sequence-to-sequence neural models, which learn from potentially noisy data and may generate ungrammatical texts.

Automatic metrics aim to be a quick and easy way to quantify how well system outputs correlate with anticipated ones, but lack the ability to provide a real estimate of the "quality" of generated outputs, which is a real concern for NLG systems. Novikova et al. [Novikova et al., 2017a] explore frequently used word-based and grammar-based metrics, finding only a weak correlation between existing automatic metrics and more detailed human judgments of system outputs.

2.5.2 Qualitative Measures

Qualitative measures involving the use of human raters is common practice in evaluating NLG output. However, there has yet to be a standard set of measures proposed, or a standard rubric for evaluating the quality of generated utterances. Table 2.6 shows some examples of criteria evaluated through human judgments in the literature. We choose different pieces of work in different domains to present a variety of different measures.

From the table, we see that not only are the criteria varied for judging system output, but the scale on which judgments are made is also variable. Additionally, a given criterion may be used to perform individual ratings, or in a ranking context.

There have been a few studies comparing related sets of automatic metrics and human judgments. Stent et al. [Stent et al., 2005] find correlations between au-

Task	Criterion	Description	Scale
Knowledge Explanation [Lester and Porter, 1997]	Coherence	Overall quality of explanations generated.	5-pt Likert
	Content	Whether the information is adequate and focused.	5-pt Likert
	Organization	How well organized the information is.	5-pt Likert
	Writing Style	Prose quality.	5-pt Likert
	Correctness	How accurate the explanations are (scientific explanations).	5-pt Likert
Weather Forecasting [Belz and Reiter, 2006]	Readability	N/A	Likert 0-5
	Clarity	N/A	Likert 0-5
	General Appropriateness	N/A	Likert 0-5
Restaurant Description [Novikova et al., 2017a]	Informativeness	Does the utterance provide all the useful information from the meaning representation?	6-pt Likert
	Naturalness	Could the utterance have been produced by a native speaker?	6-pt Likert
	Quality	How do you judge the overall quality of the utterance in terms of its grammatical correctness and fluency?	6-pt Likert
Formality Style Transfer [Rao and Tetreault, 2018]	Formality	Individually rate the formality of three sentences (source, human rewrite, and system output).	Likert -3-3
	Fluency	Individually rate the fluency of three sentences (source, human rewrite, and system output).	Likert 1-5
	Meaning Preservation	Rate the meaning similarity of a pair of sentences (combinations of source, human rewrite, and system output).	Likert 1-6
	Overall Rating	Rank the overall formality of three sentences, taking into account fluency and meaning preservation (source, human rewrite, and system output).	Ranking 3 versions

Table 2.6: Criteria and scales used in a sample of human evaluation studies.

tomatic metrics for adequacy, not but for fluency. In contrast, in their investigation of how automatic metrics correlate with human ratings in the weather forecasting domain, Reiter and Belz [Reiter and Belz, 2009] find that the human judgments they collect for clarity (Stent et al.'s “fluency”) correlate well with automatic metrics NIST, BLEU, and string edit-distance, but they find that *no* metrics correlate with human judgments for accuracy (Stent et al.'s "adequacy"). There are inherent differences in how the measures were computed (e.g. Stent et al. use single references as ground truth as opposed to Reiter and Belz's use of multiple references), but the striking differences explain how evaluations are highly subjective and potentially application-specific.

Reiter and Belz reasonably conclude that automatic metrics may be useful for evaluating linguistic quality, but that they should be used with caution, and supplemented with rigorous, well-designed human evaluation appropriate for the task at hand. Ultimately, a combination of automatic metrics and human evaluation is the standard for evaluating NLG today. In our own work on evaluation semantics and style in NNLG, we use a combination of quantitative and qualitative evaluations to explore how well our generated outputs are able to preserve content while producing diverse outputs. We present evaluations for each set of NNLG model outputs we produce in Chapter 4 and Chapter 6.

2.6 Summary

While the field of NLG has been around for decades, it has thus far primarily focused on systems that are able to faithfully reproduce the required semantics. With the rise of personal assistants, the requirements have shifted from systems that reproduce semantics, to systems that simultaneously remain faithful to content *and* are able to communicate information in a stylistically natural way.

In this chapter, we give an overview of NLG as a whole, then frame our review of literature around the data, methods, and evaluation of stylistic variation in NLG. In Section 2.2, we give an introduction to NLG both in traditional SNLG systems, and state-of-the-art end-to-end NNLG systems. We provide a detailed description of how datasets have historically been collected for the generation task, and enumerate some of the most popular datasets in NLG in Section 2.3. We then describe a few pieces of existing work on style generation, pointing out where the style gap lies, and outlining how our own work aims to fill this gap with stylistic control for NNLG. Finally, we give a detailed summary of quantitative and qualitative evaluation methods commonly used in NLG.

For the remainder of this thesis, we focus on NNLG. We begin by giving details about the motivation and design behind NNLG models, then moving on to describe how we fill the style gap by introducing data and methods for stylistic control in NNLG.

Chapter 3

Neural Natural Language Generation

3.1 Overview

The primary goal of this thesis is to address the style gap in neural Natural Language Generation (NLG) through methods for data collection and model control. In this chapter, we provide an overview of state-of-the-art data and methods in neural NLG. In subsequent chapters, we describe our own contributions, as they build on the data and methods presented here.

As we discussed in Chapter 2, neural models have recently gained immense popularity in the NLG community due to their success on related problems such as machine translation [Cho et al., 2014, Sutskever et al., 2014, Bahdanau et al., 2014]. Instead of using a traditional NLG pipeline with separate modules for sentence planning and surface realization, Neural Natural Language Generation (NNLG) models combine these steps into a single end-to-end framework, jointly learning sentence planning and surface realization from data [Dušek and Jurcicek, 2015, Mei et al., 2016, Wen et al., 2015, Mei et al., 2016, Wen et al., 2016, Nayak et al., 2017, Novikova et al., 2017b, Dusek and Jurcícek, 2016, Lampouras and Vlachos, 2016, Sharma et al., 2017]. The promise of this new paradigm is entirely data-driven generation: given a parallel dataset of meaning representations to natural language texts, a neural generator should be able to learn to generate without any rules, statistical approaches, or expensive alignment between Meaning Representation (MR)s and their outputs. The implications in different applications such as dialog systems are huge, potentially facilitating NLG systems that can be reused, and that do not have to be redeveloped for new applications [Novikova et al., 2017b].

The earliest works on end-to-end NLG used small, delexicalized datasets, such as BAGEL (around 400 utterances [Mairesse et al., 2010]), RoboCup (around 2k utterances, [Chen and Mooney, 2008]), or SF hotels and SF restaurants (each around 5k utterances, [Wen et al., 2015]). While these datasets have allowed for some exploration of End-to-End (E2E) generation, the richness of the language generated is limited: larger and more diverse datasets are needed to provide models with enough examples of complex language phenomena to allow for more diverse generation [Walker et al., 2004, Stent et al., 2004, Demberg and Moore, 2006, Rieser and Lemon, 2010].

An important attempt to address this gap came from Novikova et al. [Novikova et al., 2017b], with the E2E Generation Challenge,[1] a shared task aimed at end-to-end NNLG from structured inputs. The task presented participants with a newly-collected crowdsourced dataset of 50k utterances in the restaurant domain, significantly larger than any dataset that was currently available for NNLG, and

[1] http://www.macs.hw.ac.uk/InteractionLab/E2E/

designed to include more varied language than previous datasets. The goal was to design an end-to-end system that could take in a structured input and produce natural language outputs that were: (1) similar to gold-standard outputs written by humans as judged by automatic metrics, and (2) highly-rated by human judges for quality and naturalness. The challenge elicited a great deal of interest in the task of E2E generation from structured input without semantic alignment to outputs, and on varied evaluation metrics at scale [Novikova et al., 2017b], with the top submissions using neural models.

In this chapter, we describe in detail the task of neural NLG, using data and methods from the E2E challenge as a running example. We begin with a description of the dataset collected in Section 3.2, followed by an architectural overview of a state-of-the-art model for E2E neural NLG in Section 3.3, which was the most popular with systems that participated in the challenge. In Section 3.4, we give an overview of the evaluation metrics used in the challenge, which are widely used to evaluate state-of-the-art NNLG systems in current work. Finally, in Section 3.5, we lay down the foundations of the style gap that we will subsequently address in this thesis.

3.2 The E2E Dataset

As we have described in previous chapters, there has been much previous work on generation from structured input representations, in both Statistical Natural Language Generation (SNLG) and NNLG settings [Mairesse et al., 2010, Chen and Mooney, 2008, Dušek and Jurcicek, 2015, Mei et al., 2016, Wen et al., 2015, Mei et al., 2016, Wen et al., 2016, Nayak et al., 2017, Novikova et al., 2017b, Dusek and

Jurcícek, 2016, Lampouras and Vlachos, 2016, Sharma et al., 2017]. Generation from structured input allows for controllable generation, for example in the context of a dialog system, where we must realize actions given predetermined content.

The E2E dataset was designed with the requirements of a dialog system that provides restaurant descriptions in mind. In this case, the input for a given system response is represented as an MR, consisting of a dialog act with a set of attribute-value pairs representing the content [Novikova et al., 2016]. Many of the attributes in the E2E NLG dataset are shared with previous work on NLG in the restaurant domain, such as *food* and *area* [Mairesse et al., 2010, Wen et al., 2015]. Specifically, there are eight different attributes, each with at least two values, as shown in Table 3.1.

Attribute	Value
Name	{*Cocum, The Eagle...*}
Area	{*city centre, riverside, ...* }
Food	{*English, French, Italian, ...*}
Eat-type	{*coffee shop, restaurant, pub, ...*}
Family-friendly	{*yes* | *no*}
Near	{*Raja Indian Cuisine, Market Square, ...*}
Price-range	{*£20-25*, cheap, expensive, ... }
Rating	{*high, 1 out of 5, 3 out of 5, ...* }

Table 3.1: Attributes and sample values from the E2E dataset [Novikova et al., 2017b].

The dataset was collected by presenting the MRs to crowd workers, who are asked to write a natural language utterance that expresses the items in the MR. In order to elicit more varied and natural responses, Novikova et al. present the MRs in a pictorial format instead of as text, as shown in Figure 3.1, which could potentially represent the MR: *name[Loch Fyne], cuisine[Japanese], familyFriendly[yes] eatType[restaurant], priceRange[cheap]* [Novikova et al., 2016, Dusek et al., 2019].

The idea here is that when the crowd workers are not presented a textual table, like in Table 3.1, with specific values which necessarily suggest particular lexical items to use in their realizations, they are more likely to do more interesting content ordering and generate more diverse responses [Wang et al., 2012, Novikova et al., 2016].

Each MR in the E2E dataset has an average of 8 corresponding crowd-sourced natural language realizations, which serve as gold-standard human references [Novikova et al., 2016], and the data distribution across splits is shown in Table 3.2. The test instances are unseen in training and development even after delexicalization of restaurant names [Novikova et al., 2017b, Dusek et al., 2019].

Table 3.3 shows example MRs from the dataset for different MR sizes. A detailed comparison of the final E2E dataset, compared to SF Restaurants [Wen et al., 2015] and BAGEL [Mairesse et al., 2010] by Dusek, Novikova, and Rieser [Dusek et al., 2019] highlights some interesting improvements: for example, E2E has up to 6 sentences (an average of 1.54) per Natural Language (NL) realization, as compared to around 1 or 2 (average below 1.1) for the other sets, and the vocabulary is significantly larger (2.7k tokens for E2E, 1.2k for SF Restaurants, and 601 for BAGEL). An example of variation in the data is clear in the first MR in Table 3.3, which shows the different verbs used for expressing the relation of restaurant name to food type, i.e. "offers", "provides", and "is known for".

Split	# References	# MRs
Train	42,064	4,862
Dev	4,672	547
Test	4,694	630

Table 3.2: Data split in the E2E dataset (8 refs per MR on average).

Figure 3.1: Sample picture used for crowdsourcing E2E NLG [Novikova et al., 2016].

MR	Sample References
name[The Punter], food[Indian], priceRange[cheap]	The Punter offers cheap Indian food.
	The Punter provides Indian food in the cheap price range.
	The Punter is known for serving cheap, tasty Indian food.
name[Wildwood], eatType[pub], food[Fast food], priceRange[more than 30], customer rating[high]	Wildwood Pub, is a good place that offers burgers.
	Wildwood is a pub that is above the average price range, that serves Fast food, it has a high customer rating.
	Wildwood is a highly rated Fast-food pub, with a price range of more than 30.
name[Browns Cambridge], eatType[coffee shop], food[French], customer rating[5 out of 5], area[city centre], familyFriendly[no], near[Crowne Plaza Hotel]	Browns Cambridge has excellent ratings and is a French, coffee shop in the city centre, near Crowne Plaza Hotel.
	There is a French, coffee shop with high ratings in the city centre, near Crowne Plaza Hotel. The name is Browns Cambridge. It's not family-friendly though.
	Located near the Crowne Plaza Hotel in the city centre area, Browns Cambridge is a French food coffee shop with a rating of 5 out of 5 and is not family-friendly.

Table 3.3: Sample MRs and corresponding NL references from E2E [Novikova et al., 2017b].

3.3 The Seq2Seq Model for Neural NLG

The state-of-the-art for neural NLG is a Sequence-to-Sequence (seq2seq) encoder-decoder architecture, commonly used for related tasks such as machine translation [Bahdanau et al., 2014, Cho et al., 2014, Sutskever et al., 2014]. There are many open-source frameworks implementing the seq2seq model, including OpenNMT [Klein et al., 2018], Tensor2Tensor [Vaswani et al., 2018], and FairSeq [Gehring et al., 2017].

The popularity of the seq2seq model for language generation is clear even from just analyzing the methods used for E2E challenge submissions: 13 of the 21 systems submitted a seq2seq-based system, and both the baseline and all 8 top-scoring systems were seq2seq-based [Novikova et al., 2017b]. For reference, 2 of the remaining systems used other data-driven methods (non-seq2seq), 3 were template-based, and 2 were primarily rule-based.

In this section, we give an overview of the seq2seq architecture used by state-of-the-art neural models for NLG, using the open-source E2E baseline model, TGen,[2] as an example [Dušek and Jurcicek, 2015, Dusek and Jurcícek, 2016, Dusek, 2017]. We note that TGen was the best performing in the E2E challenge in terms an average of all quantitative metrics [Dusek et al., 2019] (which we describe in detail in Section 3.4), and thus we use it as the basis for our own style experiments in Chapter 4.

The general idea behind the seq2seq model for any application is a simple *encoder-decoder* framework: a sequence is input token-by-token to an *encoder* Recurrent Neural Network (RNN) [Rumelhart et al., 1986, Goodfellow et al., 2016] (commonly a Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] in the case of NNLG), which encodes it into a hidden state that is consumed by a *decoder* RNN to produce an output sequence, also a token at a time [Dusek, 2017].

The full seq2seq model architecture is shown in Figure 3.2, which we ref-

[2] https://github.com/UFAL-DSG/tgen

erence in our discussion. To describe the full model, we first give an outline of the input representation for converting an MR into a sequence to be consumed by an input encoder. Then, we move on to describe the core seq2seq generator, and finally conclude with details of methods for output reranking to produce a final NL realization.

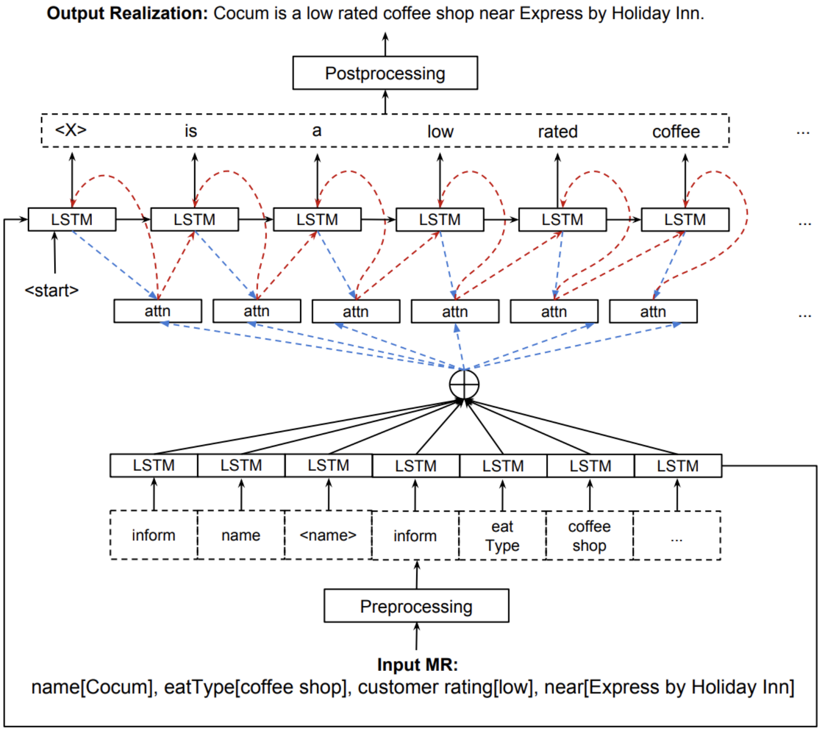


Figure 3.2: Model architecture overview for seq2seq generation, modeled around the E2E challenge baseline model [Dušek and Jurcicek, 2015, Dusek and Jurcícek, 2016].

3.3.1 Input Representation

As in the problems of machine translation and speech recognition, natural language generation from an input meaning representation is an inherently sequential problem, where the underlying question is: *How do we map sequences of tokens* *in the input (some representation of content) into a sequence of tokens in the output (natural language string)?*

In the standard seq2seq architecture [Sutskever et al., 2014, Bahdanau et al., 2014], the probability of a target output sequence $w_{1:T}$ given a source input sentence $x_{1:S}$ is modeled as shown in Equation 3.1 [Klein et al., 2018]:

$$p(w_{1:T}|x) = \prod_{1}^{T} p(w_t|w_{1:t-1}, x) \tag{3.1}$$

In the case of generation from an input meaning representation, 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 representation of an attribute-value pair in the MR. Additionally, in the context of a language generation module for a dialog system, for example, there may be additional information to encode, such as a communicative goal for the instance.

Thus, we need to represent a given input $x_{1:S}$ as a sequence of attribute-value pairs from an input MR. For example, consider the MR and NL from the E2E challenge training set shown below, which consists of a single dialog act, *inform*, and a set of 4 attribute-value pairs defining the restaurant name, type, rating, and what it is near.

$$\begin{aligned} &\text{inform(name[Cocum], eatType[coffee shop],} \\ &\text{customerRating[low], near[Express by Holiday Inn])} \end{aligned} \tag{3.2}$$

Cocum is a low rated coffee shop near Express by Holiday Inn.

A common method for MR encoding, and specifically, that used in the E2E baseline model [Dusek and Jurcícek, 2016], is to represent the MR as a sequence of attributes and value tokens, repeating shared information throughout, as shown below in (3.3) with the *inform* dialog act. While this is not the only way to represent shared information, as we show in our experiments in Sections 4.3 and 6.3, this method has been shown to work effectively [Dušek and Jurcicek, 2015, Dusek and Jurcícek, 2016].

Several steps are commonly taken in terms of preprocessing to prepare the input MR for use in the seq2seq system, as depicted at the bottom of Figure 3.2. For example, to avoid unnecessary data sparsity, lowercasing and delexicalizing inputs and outputs is a common preprocessing step in data-to-text NLG. In the case of the E2E baseline system, values for the *name* and *near* attributes are delexicalized by default, since they define "non-enumerable" inputs (i.e. they can take on any string value) [Dusek, 2017]. Note how multi-word values such as "coffee shop" and "Express by Holiday Inn", are merged into single-word tokens such that they get encoded as a single token, and also how content order of the input MR is not identical to that of the NL realization. Additionally, plural words in the NL are represented using a special "-s" token, rather than encoding the singular and plural forms as separate words [Dusek, 2017].

$$\begin{gathered}\text{inform name x-name inform eattype coffee-shop}\\ \text{inform customerrating low inform near x-near}\\ \textit{x-name is a low rated coffee-shop near x-near.}\end{gathered} \tag{3.3}$$

Given this representation for the input MR as a string sequence of triples, the next step is to convert them to a standard form for input into the sequential

RNN encoder. Tokens are traditionally represented by an embedding, or a vector of real-valued numbers [Bengio et al., 2003, Dusek, 2017]. In the case of the E2E baseline model, at training time, each token in the inputs (i.e. both MR and NL for each training instance pair) is given an integer ID, which maps to a vector in an embedding matrix which is randomly initialized and then learned during training [Dusek, 2017].

3.3.2 Seq2Seq Generation

Encoding

The core generator from the E2E baseline is based on the seq2seq model with attention from Bahdanau et al. [Bahdanau et al., 2014]. In the general seq2seq model, an RNN is used for internal encoding to convert the input sequence $x_{1:S}$ into internal states (specifically, hidden states and internal states) to be consumed in the network [Dusek, 2017]. The RNN unit [Rumelhart et al., 1986, Goodfellow et al., 2016] is a simple class of neural network designed to process sequential data, to predict next-token probabilities based on *all previous tokens*. In the standard RNN architecture, the repeating cell that allows this is usually a single layer *tanh* unit. The RNN cell theoretically allows for long-distance dependencies, but in practice, does not work well [Hochreiter and Schmidhuber, 1997, Bengio et al., 1994].

In the specific case of the variable-sized (often long) inputs for text generation, the preferred RNN unit is the LSTM cell [Hochreiter and Schmidhuber, 1997], since it does a much better job at modeling long-term dependencies. Thus the recurrence equation used for encoding input sequence x in terms of the hidden states h and internal cell states C is shown in Equation 3.4, with h_0 and C_0 initially set to 0 [Dusek, 2017]:

$$(h_t, C_t) = \text{lstm}(x_t, h_{t-1}, C_{t-1}) \tag{3.4}$$

To perform the encoding, the LSTM unit itself is governed by four network units, commonly referred to as the cell, input gate, output gate, and forget gate (as opposed to the single layer in a standard RNN). Each gate is composed of a sigmoid layer that outputs a value between 0-1, dictating whether or not information passes through (0 means none of the information, 1 means all). Note that learned weight matrices 3.5-3.11 are denoted throughout by W, bias terms are denoted by b, and $\circ$ denotes a concatenation operation [Dusek, 2017].

First, the forget gate f in Equation 3.5 decides how much information to pass on to the next cell state. This is dictated by the current input x_t and previous hidden state h_{t-1} for each part of the previous cell state C_{t-1} as in the recurrence in Equation 3.4.

$$f_t = \sigma(W_f(h_{t-1} \circ x_t) + b_f) \tag{3.5}$$

The next step controls how much information is stored in the new cell state, C_t, as in Equation 3.6. To do this, the input gate i_t must decide which values to update based on the new set of candidate values, $\widetilde{C}_t$, generated with a *tanh* layer.

$$\begin{aligned} i_t &= \sigma(W_i(h_{t-1} \circ x_t) + b_i) \\ \widetilde{C}_t &= \tanh(W_f(h_{t-1} \circ x_t) + b_f) \end{aligned} \tag{3.6}$$

The new cell state C_t in Equation 3.7 is thus updated based on the product of the old cell state C_{t-1} and the forget state f_t, and on adding in the new information from the input cell i_t, which is shaped by the new candidate values $\widetilde{C}_t$.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t \tag{3.7}$$

At the last stage, we determine our final hidden output at the current timestep h_t based on a filtered version of the cell state using a sigmoid layer to generate o_t, scaled with the *tanh* in Equation 3.11.

$$\begin{gathered} o_t = \sigma(W_o(h_{t-1} \circ x_t) + b_t) \\ h_t = o_t \cdot \tanh c_t \end{gathered} \tag{3.8}$$

Decoding

Given the fully encoded sequence, the second stage of generation is decoding, as shown in the top half of Figure 3.2. This stage uses the final encoder state C_n and the hidden outputs h to generate the target outputs sequence $w_{1:T}$, from Equation 3.1, again using an LSTM [Hochreiter and Schmidhuber, 1997]. The probability of each output token in the target sequence is defined in Equation 3.9, where s_t is the hidden unit output and c_t is the attention model (which we describe in more detail below):

$$p(w_{1:T}|x) = \text{softmax}((s_t \circ c_t)\, W_Y) \tag{3.9}$$

The hidden unit output s_t is computed as in Equation 3.10. s_t and $C_t^{'}$ begin as the outputs of the final encoder state above (i.e. $s_0 = h_n$ and $C_0^{'} = C_n$).

$$(s_t, C_t^{'}) = \text{lstm}((x_{t-1} \circ c_{t-1})W_s, W_{t-1}, C_{t-1}^{'}) \tag{3.10}$$

c_t represents the attention model, which is represented as a sum over *all encoder hidden states* and weighted by a feed-forward network that represents the alignment model α_{ti}, with v, W, and U learned weight matrices [Dusek and Jurcícek, 2016, Dusek, 2017, Bahdanau et al., 2014, Sutskever et al., 2014, Luong et al., 2015].

$$\begin{aligned} c_t &= \sum_{i=1}^{n} \alpha_{ti} h_i \\ \alpha_{ti} &= \mathrm{softmax}(v_\alpha^T \tanh(W_\alpha s_{t-1} + U_\alpha h_i)) \end{aligned} \tag{3.11}$$

Additionally, beam search [Sutskever et al., 2014, Bahdanau et al., 2014] is implemented in the decoder, where the n most probable *next* output sequences for a given timestamp are always expanded by an extra token, with the resulting n outputs used for the next decode step.

3.3.3 Output Reranking

The n-best beam outputs from the decoder are handed over to an output reranker, which uses an additional encoder and classification layer to rank the beam outputs by how well they retain the information in the original MR, effectively penalizing outputs that make common semantic errors, such as deletions, repetitions, and substitutions of values from the MR [Dusek and Jurcícek, 2016, Dusek, 2017]. The reranker in the E2E baseline model is based on Wen et al.'s reranker [Wen et al., 2015].

The reranking classifier works by first outputting a binary vector indicating the presence or absence of dialog act tags, attribute names and values in each beam output. The input MR is also converted to a similar binary vector (1-hot encoding) for comparison with each beam output, and the reranking penalty is the weighted

Hamming distance between the classification output of the reranking classifier with the binary MR sequence. Next, the sentences with the highest scores are chosen and passed to a n-gram ranker which maximizes the BLEU score[3] between the beam and MR binary vectors. The ranked output with the highest score is ultimately picked as the final output sentence [Dusek, 2017].

Given the final output sequence, the final step at the top of Figure 3.2 is the relexicalization of the output sequence for *name* and *near* attributes removed in preprocessing, thus returning an NL string as the target output from the model for the input MR.

3.4 Evaluation Methods

To introduce the evaluation metrics used to evaluate systems in the E2E challenge (and commonly used to evaluate recent work on NNLG in general), we revisit the challenge goal: to design an end-to-end system that could take in a structured input and produce natural language outputs that were: (1) similar to gold-standard outputs written by humans as judged by automatic metrics, and (2) highly-rated by human judges for quality and naturalness. We note here that the seq2seq model we described in the previous section, based on the baseline system in the challenge, was the most popular method, with over 60% submissions (13 out of 21) using a seq2seq model [Dusek et al., 2019].

[3] We define BLEU and other automatic metrics in Section 3.4.

3.4.1 Quantitative Evaluation

The E2E challenge evaluation included the use of automated metrics for judging system output and ranking competing systems, including providing systems with an automatic evaluation script for the following metrics, given a set of system outputs and the corresponding references.[4]

The metrics used were: BLEU (n-gram precision), NIST (weighted n-gram precision), METEOR (n-grams with synonym recall), and ROUGE (n-gram recall), and CIDEr (TF-IDF weighted n-gram cosine similarity), which are commonly used in the machine translation community (as described in Section 2.5.1).

Table 3.4 shows the scores of the TGen baseline and the top 5 systems in terms of a normalized average of the automatic metrics (last column). The TGen baseline was still the highest scoring of all system in terms of the normalized average, but the top systems frequently outperformed TGen for specific metrics (all scores that outperform the baseline are shown in bold). We note here that all 5 top performing systems were seq2seq based. The top performing submission and overall challenge winner [Dusek et al., 2019], SLUG [Juraska et al., 2018], used a TGen-based seq2seq architecture with MR classification and reranking. TNT1 [Oraby et al., 2018c] and TNT2 [Tandon et al., 2018], both resulting from work in this thesis, were also TGen-based, using synthetic data augmentation and MR data shuffling techniques, respectively. NLE [Agarwal et al., 2018] was a char-based seq2seq model (as opposed to the word-based seq2seq in the TGen baseline), also with MR classification and reranking as with SLUG. Finally, HARV [Gehrmann et al., 2018] was a seq2seq model with reranking penalty and ensembling.

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

Model	BLEU	NIST	METEOR	ROUGE	CIDEr	Norm. Avg.
TGen Baseline [Dusek and Jurcícek, 2016]	0.6593	8.6094	0.4483	0.6850	2.2338	0.5754
SLUG [Juraska et al., 2018]	**0.6619**	**8.6130**	0.4454	0.6772	**2.2615**	0.5744
TNT1 [Oraby et al., 2018c]	0.6561	8.5105	**0.4517**	0.6839	2.2183	0.5729
NLE [Agarwal et al., 2018]	0.6534	8.5300	0.4435	0.6829	2.1539	0.5696
TNT2 [Tandon et al., 2018]	0.6502	8.5211	0.4396	**0.6853**	2.1670	0.5688
HARV [Gehrmann et al., 2018]	0.6496	8.5268	0.4386	**0.6872**	2.0850	0.5673

Table 3.4: Baseline and top 5 systems in the E2E challenge, based on a normalized average of automatic metrics [Dusek et al., 2019].

Table 3.5 shows sample outputs for these systems for 2 MRs, as compared to one of the human references collected through crowdsourcing, and to the TGen baseline output. **MR 1** in the table is a shorter example, with only 4 attribute-value pairs. From the outputs for MR 1, we see that SLUG, TNT1, TNT2, and HARV all realize all MR content, but in this case TGen actually hallucinates an additional content item, *high price range*, not seen in the MR. NLE substitutes the restaurant name *Cocum* for *Cotto*, and makes a repetition of the value *high* for attribute *customer rating* (we point out that NLE does not do any delexicalization, while TGen, SLUG, TNT1, and TNT2 all do). We also note that TGen is the only system that realizes the content in two separate sentences.

For the longer example in MR 2, TGen, SLUG, and TNT1 all output realizations with two sentences. We see in this case that TGen, SLUG, TNT1, and TNT2 make no semantic mistakes. We observe that both NLE and HARV substitute different restaurant names, and HARV omits the food value *French*.

We point out here that while the automatic metrics are useful to rank the systems and as internal validation metrics for system development [Dusek, 2017], they do not provide a transparent way to understand the *types* of mistakes the models make, i.e. the deletions, repetitions, and substitutions we see in the outputs in Table 3.5. In our own evaluations of our work in Sections 4.4.1 and 6.4.1, we introduce more detailed metrics for semantic error rates to better quantify our model errors.

MR 1	`name[Cocum], eatType[coffee shop], customer rating[high], near[Burger King]`
Human	*Near Burger King there is a highly rated coffee shop named Cocum.*
TGen	Cocum is a highly rated coffee shop with a high price range. It is located near Burger King.
SLUG	Cocum is a highly rated coffee shop near Burger King.
TNT1	Cocum is a high customer rating coffee shop near Burger King.
NLE	Cotto is a coffee shop near Burger King with a high customer rating and a high customer rating.
TNT2	Cocum is a coffee shop near Burger King with a high customer rating.
HARV	There is a highly rated coffee shop named Cocum near Burger King.

MR 2	`name[The Phoenix], eatType[pub], food[French], priceRange[high], area[riverside], familyFriendly[yes], near[Raja Indian Cuisine]`
Human	*The Phoenix is a high priced pub serving French cuisine situated on the riverside near Raja Indian Cuisine which is child friendly.*
TGen	The Phoenix is a high priced french pub near Raja Indian Cuisine in the riverside area. It is children friendly.
SLUG	The Phoenix is a French pub in the riverside area near Raja Indian Cuisine. It is child friendly and has a high price range.
TNT1	The Phoenix is a children friendly french pub near Raja Indian Cuisine in the riverside area with a high price range.
NLE	The Punter is a children friendly French pub located near Raja Indian Cuisine in the riverside area with a high price range.
TNT2	The Phoenix is a high priced french pub near Raja Indian Cuisine in the riverside area. It is child friendly.
HARV	The Wrestlers is a high priced, child friendly pub in the riverside area near Raja Indian Cuisine.

Table 3.5: Sample outputs from competing systems in the E2E challenge, compared to a single human reference and the baseline system output.

3.4.2 Qualitative Evaluation

The second type of evaluation in the E2E competition was the human evaluation, conducted by the organizers on all primary systems and the baseline model, using the CrowdFlower (now FigureEight) crowdsourcing platform.[5] The evaluation involved showing crowd workers five randomly selected system outputs and a matching human NL output for reference, and asking them to rank the outputs from best to worst (allowing ties).

The rankings were based on two separate metrics, *quality* and *naturalness*. Quality took a group of metrics, such as grammatical correctness, fluency, adequacy, which would be considered the primary metrics for direct application in a real NLG system. Crowd workers were presented the MR along with the system outputs when judging for quality (meaning that content preservation factored into quality), and asked: *"How do you judge the overall quality of the utterance in terms of its grammatical correctness, fluency, adequacy and other important factors?"* [Dusek et al., 2019]

Naturalness considered how likely it is that the output could have been produced by a native speaker. The MRs were not shown to crowd workers when judging for naturalness [Dušek et al., 2018]. They workers were explicitly asked: *"Could the utterance have been produced by a native speaker?"* [Dusek et al., 2019]

The competition evaluation results were then computed using the TrueSkill algorithm from Sakaguchi et al. [Sakaguchi et al., 2014]. For both quality and naturalness, pair-wise comparisons were first made, and then the systems were ranked by their TrueSkill scores, and then clustered, such that different systems within the

[5] `https://www.figure-eight.com/`

same cluster are considered tied. Clustering was done using bootstrap resampling ($p \leq 0.05$).[6] For quality, the top-ranking system (i.e. the only system in the 1^{st} cluster), was the SLUG system [Dusek et al., 2019], while in the naturalness evaluation, the top system was SHEFF2 [Chen et al., 2018] (again the only system in the 1^{st} cluster). Interestingly, SHEFF2 did not score in the top 5 on the automatic evaluations, in fact, it was ranked 18^{th} in terms of the normalized average automatic metric ranking. This again points to the discrepancy frequently found between automatic metrics and human evaluation frequently referenced in previous work [Novikova et al., 2017a, Reiter, 2018], as we discussed in Section 2.5.

3.5 The Style Gap in NNLG

In considering the overall findings of the E2E challenge, we draw some interesting observations about the state-of-the-art in terms of controlling semantics while producing good stylistic variation for current NLG models (particularly seq2seq models), which is the primary interest of our work in this thesis.

In their analysis of overall trends in participating systems in the E2E challenge, Dusek et al. [Dusek et al., 2019] find that seq2seq methods in general did best in terms of naturalness, with the bottom cluster containing primarily rule-based systems. Additionally, they note that systems that attempt to introduce diversity in their models are penalized on naturalness. For quality, they note that no particular architecture particularly outperformed another; in fact, seq2seq models made up most of the bottom two quality clusters.

The style gap in NLG is also clear from analyzing the outputs of the best-

[6] More detail about the evaluations can be found on the challenge homepage: http://www.macs.hw.ac.uk/InteractionLab/E2E/

performing systems in terms of automatic metrics, such as the examples we showed in Table 3.5. The model outputs were not particularly stylistically diverse, rigidly expressing the content requirements. We point to the fact that in terms of automatic metrics, increasing attempts at introducing stylistic diversity can lead to poor performance on automatic metrics (since the metrics compare to a small number of references), as it did for the naturalness evaluation.

Thus, from the E2E challenge, Dusek et al. [Dusek et al., 2019] conclude that while seq2seq models perform comparably well in terms of automatic metrics and naturalness evaluations, vanilla NNLG models do not yet have a good mechanism for semantic control, frequently making semantic errors. Additionally, they are frequently outperformed by rule-based systems in terms of diversity measures such as human quality evaluation, length, and complexity. We add to these findings that vanilla seq2seq models do not provide an explicit mechanism to control style in NNLG outputs, particularly while trying to maintain good semantics: thus leading us back to our own goals in this thesis.

3.6 Summary

In this chapter, we described the state-of-the-art in NNLG through the findings of the E2E challenge, a competition centered around new methods for end-to-end data-to-text generation. We described the novel dataset of 50k MR to NL in Section 3.2, created through a massive crowdsourcing effort and released as part of the challenge [Novikova et al., 2017b]. In Section 3.3, we described the underlying architecture behind state-of-the-art seq2seq models for neural NLG, centered around the E2E challenge baseline system, TGen. Then, we moved on to describe the quantitative and qualitative evaluation metrics used in the challenge in Section 3.4, showing us real examples of the style gap in NNLG.

While systems in the challenge were successful at generating utterances that preserved some of the semantics in the data, the limitations of NNLG systems are clear: (1) there is no obvious mechanism to control semantics or style, (2) datasets for NNLG must be large, and are thus tedious to acquire through current crowd sourcing methods. Both limitations clearly identify a gap in the state of the art, to introduce methods for model control and data curation for NNLG: thus, we begin our own attempts to fill this style gap in the remainder of this thesis.

Chapter 4

Producing Style in Neural NLG

4.1 Overview

Thus far in this thesis, we have motivated the adoption of the neural paradigm shift in Natural Language Generation (NLG), centered around the promise of entirely data-driven generation, requiring only a large corpus, and without the need for intermediate rules or representations. In Chapter 2, we ran through a streamlined history in NLG, describing the limitations of the previously popular modular Statistical Natural Language Generation (SNLG) model, based on a pipelined framework moving from content planning to surface realization. In Chapter 3, we gave an overview of the general architecture of the state-of-the-art Sequence-to-Sequence (seq2seq) model for Neural Natural Language Generation (NNLG), framed around the popular End-to-End (E2E) generation challenge, which paved the way for a vast array of work on end-to-end neural generation, including our own [Wiseman et al., 2018, Juraska et al., 2018, Juraska and Walker, 2018, Oraby et al., 2018a, Oraby et al., 2018b, Oraby et al., 2018c, Reed et al., 2018, Tandon et al., 2018, Jagfeld et al., 2018].

Our exploration of the state-of-the-art NNLG architecture and the outputs they are able to produce, for example in the context of the E2E challenge we covered in Chapter 3, leads us to the conclusion that there is much work to be done to allow for truly controllable NNLG models. We have seen that neural models are notorious for making semantic errors such as deleting, repeating or hallucinating content; in fact, several pieces of work, including the E2E challenge itself, have centered their NNLG evaluations at least partly on automatic metrics aimed at quantifying these types of mistakes, and have focused their efforts on semantic fidelity [Dusek and Jurcícek, 2016, Lampouras and Vlachos, 2016, Mei et al., 2016, Wen et al., 2015]. We have also seen why it is commonly suggested that neural models employ a “frequentist” approach: learning only the simplest and most prevalent way to realize the required content from training, resulting in outputs that are dull and repetitive in structure.

In evaluating findings and lessons learned from the E2E challenge, Dusek et al. [Dusek et al., 2019] conclude that although seq2seq models (most popular in the E2E challenge) score comparably well on automatic metrics, they have notable shortcomings: (1) they make frequent mistakes on expressing the required content from a Meaning Representation (MR), (2) they “lack a strong semantic control mechanism applied during decoding”, and (3) they are generally outperformed by non-seq2seq models in terms of quality metrics such as complexity, length, and output diversity. We add to these observations the lack of any explicit mechanism to control *stylistic* diversity.

What we want to achieve in this thesis, consequently, are models that can produce output that both satisfies the required semantics as defined by an input MR, *and* simultaneously includes interesting and diverse structural and stylistic constructions. In this chapter, we describe our efforts to produce the first instance of controllable NNLG.

First, we reason that in order to measure semantic and stylistic fidelity in model outputs, we must be able to characterize both semantic and stylistic choices in our inputs. Thus, to do this, we decide that we need: (1) training data where we *know* both semantic and stylistic properties for each instance at train time, and (2) a mechanism for introducing some stylistic supervision to the model, dictating the choices we want the model to make at generation time.

We describe our efforts to obtain training data for our first requirement in Section 4.2, beginning with the E2E challenge corpus [Novikova et al., 2017b], which we described in detail in Chapter 3. As is, the E2E dataset satisfies half of our training data requirement: we know the semantics associated with each instance, but we need a way to introduce some simple stylistic variation into the instances to allow our models to learn different ways of expressing the same content.

To create this setting for controllable NNLG, we use the PERSONAGE statistical generator, which takes as input a representation of content, and outputs different ways to express that content, based on a set of rules defining Big-Five personalities [Mairesse and Walker, 2007]. These rules govern decisions such as word choice, pragmatic marker insertion, and aggregation operations. We sample a set of unique MRs from the E2E Challenge training set, and using PERSONAGE, we create 5 variations for each MR, each one based on a different personality. In this way, we synthetically design the PERSONAGENLG corpus: a set of 88k MR to Natural

Language (NL) utterances in five different personality styles. This corpus provides us with a controlled environment for testing whether an NNLG model can learn to produce both the required content *and* style for a given instance, where personality is a proxy for a multitude of different style choices. The fact that the PersonageNLG dataset is synthetically generated means that the stylistic variation in the corpus is limited to what we can control; thus, while the PersonageNLG corpus is not as natural as human crowdsourced data, it provides a perfect experimentally controlled environment for our first experiments on controllable NNLG.

Given the PersonageNLG corpus, we experiment with the second requirement for controllable NNLG in Section 4.3: a mechanism for introducing stylistic supervision to the model. We try two different methods for supervision, one providing only a single token identifying the personality type of the instance, and one with a more detailed encoding of the stylistic choices made in the instance, dictating more explicitly what choices the model should make, such as whether to include a particular hedge or pragmatic marker. We compare each supervision method to a vanilla model that does not use any style encoding.

In Sections 4.4.1 and 4.4.2, we present a set of quantitative and qualitative evaluation metrics aimed at measuring the semantic and stylistic quality of our generated outputs for each model. We include automatic metrics following the standards set by the E2E challenge to evaluate competing systems, but also introduce our own metrics to provide a more detailed analysis of the *types* of errors each model makes, and what stylistic choices it is able to produce on demand. We find that while models with less supervision make the fewest semantic errors, they lose any distinctive stylistic variation; with our most supervised model, however, we are able to achieve our goal: we can *both* produce stylistically varied outputs that correlate with the required personalities, *and* preserve semantic fidelity with notably few errors [Oraby et al., 2018b, Oraby et al., 2018a].

Finally, in Section 4.5, we present experiments aimed at generalizing from what our models have seen in training: we train our model as before on single Big-Five personalities, but required at generation time that the model generates output that is a *combination* of two personalities. Our results show that the models are able to produce novel outputs that appear to extrapolate between the two required personalities.

4.2 The PersonageNLG Corpus

As the first step towards our goal of training models that are capable of producing stylistically varied outputs given an input MR, we need training data where we *know* both semantic and stylistic properties for each instance, and where the stylistic properties are well-defined, enumerable, and measurable. In this section, we describe how we augment the E2E challenge dataset with synthetically generated variations from the PERSONAGE generator [Mairesse and Walker, 2010] to produce a dataset of known semantics and stylistic variations to train NNLG models to produce style.

4.2.1 Personage: A Statistical NLG Engine

The PERSONAGE (*"PERSONAlity GEnerator"*) NLG engine [Mairesse and Walker, 2010] is a statistical natural language generator that converts a dialog act and meaning representation into a natural language string that expresses the content in one of the Big Five personality types [McAdams and Pals, 2006], based on parameters from the psychology literature defining personality-specific lexical choices or syntactic structures [Mehl et al., 2006, Oberlander and Gill, 2006, Pennebaker and King, 1999, Thorne, 1987, Mairesse and Walker, 2010].

The architecture of the Personage generator is shown in Figure 4.1, based around the pipelined SNLG architecture we described in Section 2.2, with separate modules for content planning, sentence planning, and surface realization. As shown in the sentence planning module, there are different types of style choices that Personage can produce, namely the choice of syntactic template, aggregation operations, pragmatic marker usage, and lexical choice. Personage requires as input: (1) a high-level communicative goal, (2) an MR defining the content to express, and (3) a parameter file that tells it how frequently to use each of its stylistic parameters in the sentence planning phase.

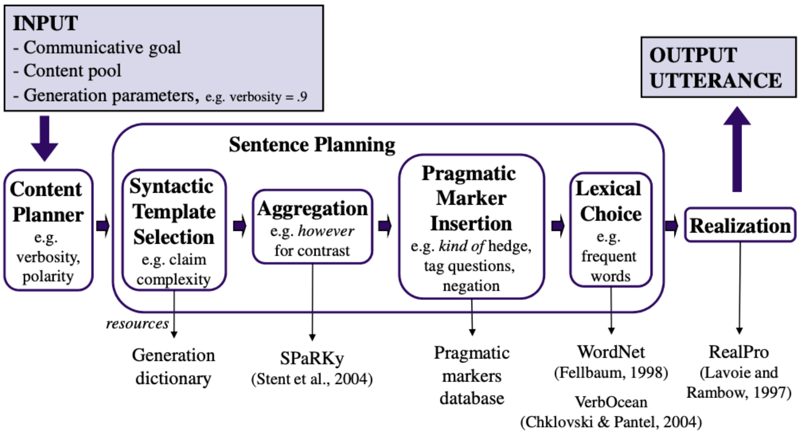

INPUT
- Communicative goal
- Content pool
- Generation parameters, e.g. verbosity = .9
OUTPUT UTTERANCE
Sentence Planning
Content Planner
e.g. verbosity, polarity
Syntactic Template Selection
e.g. claim complexity
Aggregation
e.g. however for contrast
Pragmatic Marker Insertion
e.g. kind of hedge, tag questions, negation
Lexical Choice
e.g. frequent words
Realization
resources
Generation dictionary
SPaRKy (Stent et al., 2004)
Pragmatic markers database
WordNet (Fellbaum, 1998)
VerbOcean (Chklovski & Pantel, 2004)
RealPro (Lavoie and Rambow, 1997)

Figure 4.1: The Personage generator architecture [Mairesse and Walker, 2010].

In the first phase of sentence planning, syntactic template selection, Personage looks at a stored, hand-crafted "generation dictionary", which contains a set of DSyntS (*"Deep SYNTactic Structures")* [Mel'cuk, 1988] defining syntactic templates that can be filled with variable content items from the input MR. The way these templates are combined and the choice of words to be used are then controlled by the rest of the sentence planning pipeline, and the final resulting (filled) DSynt is then converted into an NL string using the popular off-the-shelf surface realizer, RealPro [Lavoie and Rambow, 1997].

To illustrate the different personality-based variations that Personage is capable of generating given an input dialog act and MR, we focus our discussion here around an example MR from the E2E corpus. Table 4.1 shows the example E2E MR and 6 different variations generated using Personage: one with no aggregation or pragmatic markers (i.e. each attribute in its own sentence), and one for each of 5 different Big-Five personalities, Agreeable, Disagreeable, Conscientious, Unconscientious, and Extravert.

We note that here we use a subset of the different sentence planning operators and personalities that Personage can produce to focus our discussion on easily identifiable stylistic choices and prominent personality types [Oraby et al., 2018b]. We describe the specific parameters defining aggregation and pragmatic marker choices governing some of the variations Personage is able to produce in the following sections.

Aggregation in Personage

Aggregation is the aspect of sentence planning that combines one or more distinct propositions in a complex sentence, for example defining whether to create

Personalities	**Realization**
E2E MR	NAME[FITZBILLIES], EATTYPE[PUB], FOOD[ITALIAN], PRICERANGE[MODERATE], CUSTOMER RATING[DECENT], AREA[RIVERSIDE], FAMILYFRIENDLY[NO],NEAR["THE SORRENTO"]
NO AGGREGATION/ NO PRAGMATIC MARKERS	Fitzbillies is a pub. Fitzbillies has a decent rating. Fitzbillies is moderately priced. Fitzbillies is in riverside. Fitzbillies is an Italian restaurant. Fitzbillies is not family friendly. Fitzbillies is near The Sorrento.
AGREEABLE	Let's see what we can find on Fitzbillies. I see, well it is a pub with a decent rating, also it is an Italian restaurant in riverside and moderately priced near The Sorrento, also it isn't family friendly, you see?
DISAGREEABLE	I mean, everybody knows that moderately priced Fitzbillies is in riverside with a decent rating. It's near The Sorrento. It isn't family friendly. It is an Italian place. It is a pub.
CONSCIENTIOUS	Let's see what we can find on Fitzbillies. I see, well it is a pub with a decent rating, it isn't kid friendly and it's moderately priced near The Sorrento and an Italian restaurant in riverside.
UNCONSCIENTIOUS	Oh god yeah, I don't know. Fitzbillies is a pub with a decent rating, also it is moderately priced near The Sorrento and an Italian place in riverside and it isn't kid friendly.
EXTRAVERT	Basically, Fitzbillies is an Italian place near The Sorrento and actually moderately priced in riverside, it has a decent rating, it isn't kid friendly and it's a pub, you know.

Table 4.1: Personage personality-based variations for an E2E MR.

complex sentences by combining attributes into phrases and what types of combination operations are used.

Table 4.2 illustrates some of the sentence planning operations produced by PERSONAGE for aggregation in different personalities. The first column of the table defines the operator itself, followed by an example usage, and finally a parameter setting defining the use of that operator in each of the five personalities we are interested in. Each parameter value in PERSONAGE can be set to `high`, `low`, or `mid` (effectively "`don't care`"). Operations with values of mid may occur in the output but they are not indicative of the trait.

For example, the AGREEABLE column in Table 4.2 shows that for the extravert personality, PERSONAGE is set to use conjunctions, the "also" cue word, and ellipses frequently, but to use separation into separate sentences (period aggregation) infrequently. The rest of the aggregation parameters, “with” cue word and merge, are not uniquely indicative of the Agreeable personality (indicated by a value of “mid”). Note also that some personalities are not characterized at all by aggregation operators, such as the Conscientious and Unconscientious personalities, who have “mid” values for all aggregation operations.

Operator	**Example**	**Agree.**	**Disag.**	**Consc.**	**Uncon.**	**Extra.**
Period	*X serves Y food. It is in Z.*	low	high	mid	mid	low
“With” cue	*X is in Y, with Z.*	mid	mid	mid	mid	low
Conjunction	*X is Y and it is Z.*	high	low	mid	mid	high
Merge	*X is Y and Z*	mid	mid	mid	mid	mid
“Also” cue	*X has great Y, also it has nice Z.*	high	mid	mid	mid	high
Ellipsis	*X has ... it has great Y*	high	mid	mid	mid	high

Table 4.2: Aggregation operations produced by Personage for different personalities [Mairesse and Walker, 2010].

To see the effect of the aggregation operators on each personality variation, cross-reference the aggregation operations in Table 4.2 with an examination of the outputs in Table 4.1. The simplest choice for aggregation does not combine attributes at all: this is represented by the period operator, which, if used persistently, results in an output with each content item in its own sentence as in the No Aggregation/No Pragmatic Markers row, or the content being realized over multiple sentences as in the disagreeable row (5 sentences). However, if the other aggregation operations have a high value, personage prefers to combine simple sentences into complex ones whenever it can, e.g., the extravert personality example in Table 4.1 combines all the attributes into a single sentence by repeated use of the ALL MERGE and CONJUNCTION operations. The CONSCIENTIOUS row in Table 4.1 illustrates the use of the WITH-CUE aggregation operation, e.g., *with a decent rating*. Both the AGREEABLE and CONSCIENTIOUS rows in Table 4.1 provide examples of the ALSO-CUE aggregation operation.

Pragmatic Markers in Personage

The pragmatic operators we focus on in PERSONAGE are shown in Table 4.3. In general, the pragmatic markers are divided into eight categories (as shown by the grouping of rows in Table 4.3 and marked by abbreviated operator names): *acknowledgements (ack), competence mitigations, downtoners or softener hedges (down), emphasizers (emph), in group markers, expletives and near expletives, initial rejections (init reject), request confirmations,* and *tag questions*.

These pragmatic operators are intended to achieve particular pragmatic effects in the generated outputs: for example the use of a downtoner hedge such as *sort of* softens a claim and affects perceptions of friendliness and politeness [Brown and Levinson, 1987], while the exaggeration associated with emphasizers like *actually, basically, really* influences perceptions of extraversion and enthusiasm [Oberlander and Gill, 2004, Dewaele and Furnham, 1999]. In PERSONAGE, the pragmatic parameters are attached to the syntactic tree at *insertion points* defined by syntactic constraints, e.g., *emphasizers* are adverbs that can occur sentence initially or before a scalar adjective.

Each personality model uses a variety of pragmatic parameters, as can be seen by cross-referencing the markers in Table 4.3 with the personality variations in Table 4.1. For example, the AGREEABLE personality is characterized by frequent use of *acknowledgements*, with *i see, well* in the realization, as well as the use of *request confirmation let's see* and *tag questions*, e.g. *you see?*. Disagreeable, on the other hand, does not use *acknowledgements*, but instead frequently uses *competence mitigation*, e.g. *everybody knows.* A personality like Unconscientious, which was not characterized by the use of any unique aggregation operations, is more clearly defined by the use of pragmatic operations such as *expletives*, e.g. *Oh god*, and *initial rejections*, e.g. *I don't know.*

Operator	**Example**	**Agree.**	**Disag.**	**Consc.**	**Uncon.**	**Extra.**
ACK_I_SEE	*I see*	high	low	high	low	low
ACK_RIGHT	*right*	high	low	mid	mid	low
ACK_WELL	*well*	high	low	high	low	low
ACK_YEAH	*yeah*	high	low	low	high	high
COMPETENCE MITIG.	*obviously, everybody knows*	low	high	mid	mid	mid
DOWN_ERR	*err*	mid	mid	low	high	low
DOWN_I_MEAN	*I mean*	mid	mid	low	high	low
DOWN_LIKE	*like*	mid	mid	low	high	high
DOWN_SORT_OF	*sort of*	high	low	high	low	low
DOWN_SUBORD	*I think that, I guess*	high	low	high	low	low
EMPH_BASICALLY	*basically*	low	high	mid	mid	high
EMPH_EXCLAIM	*!*	mid	mid	low	high	high
IN GROUP MARKER	*pal, mate*	high	low	low	high	high
EXPLETIVES	*oh god, damn*	low	high	low	high	mid
NEAR EXPLETIVES	*oh gosh, darn*	mid	mid	low	high	mid
INIT REJECT	*I don't know, I am not sure*	low	high	low	high	low
REQUEST CONFIRM.	*Let's see, Did you say?*	high	low	high	low	high
TAG QUESTION	*alright?, you see? ok?*	high	low	mid	mid	high

Table 4.3: Pragmatic markers produced by Personage for different personalities [Mairesse and Walker, 2010].

4.2.2 Corpus Creation

There is a long tradition in the artificial intelligence community of using slightly synthetic tasks and datasets in order to test the ability of particular models to achieve these tasks, and some recent work has used this approach in various applications [Weston et al., 2015, Dodge et al., 2015]. In reference to our own goals, to evaluate whether a neural model is capable of producing required stylistic choices while preserving semantics, we need to have a finite set of well-defined style choices for the model to learn, and we need to have *enough* of them such that they are not washed out by the language model. Thus, to generate a corpus that is semantically grounded and also includes marked stylistic choices, we utilize MRs from the E2E challenge, and synthetically generate novel realizations in different personality styles using PERSONAGE, resulting in the PERSONAGENLG corpus for stylistic variation in NNLG. We describe the details of our corpus creation in this section.

We begin by replicating the E2E Generation Challenge setup, preserving the split between the train, development, and test sets, since the dataset ensures that no development or test set MRs are in the training set. The frequencies of longer utterances (MRs with more attributes) vary across train and test, with a plot of actual distributions in Figure 4.2, showing how the test set was designed to be challenging, as compared to other datasets in the restaurant domain such as SF RESTAURANTS [Wen et al., 2015], which averages less than two attributes per MR [Novikova et al., 2017b]. The combined training and development sets from E2E include 3,784 unique MRs, and test contains 278 unique MRs.

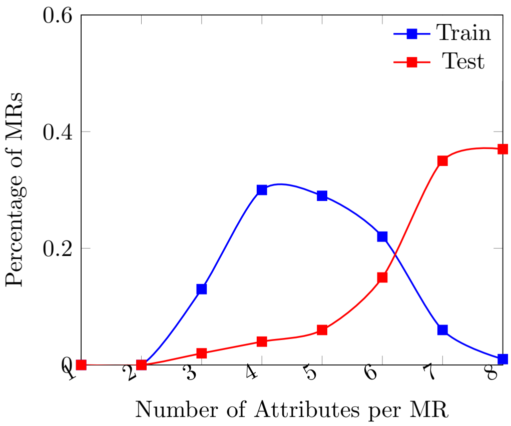


Figure 4.2: MR distribution in PersonageNLG train.

Given the unique input MRs for each split, we then set out to use Personage to create training data mapping the same MR to multiple personality-based variants. We use the values set for the aggregation operations in Table 4.2 and pragmatic markers in Table 4.3 using the stylistic models defined by [Mairesse and Walker, 2010] our five prominent Big-Five personalities, agreeable, disagreeable, conscientious, unconscientious, and extravert. We use the Personage generator mostly off-the-shelf, only needing to add in simple DSyntS for novel E2E attributes that did not exist in the original Personage dictionaries [Mel'cuk, 1988, Lavoie and Rambow, 1997, Mairesse and Walker, 2010], such as *near* (nearby establishments).

For each unique MR in the data, we generate realizations for each personality: specifically, we generate around 17.7k variations per personality, resulting in a training set of around 88k MR to NL instances, and a test set of around 1.3k instances (one per personality). Table 4.4 shows the precise distribution of data in the corpus.[1] We note that while Personage can produce 10's of variations for each

[1] The PersonageNLG corpus is available at: https://nlds.soe.ucsc.edu/stylistic-variation-nlg

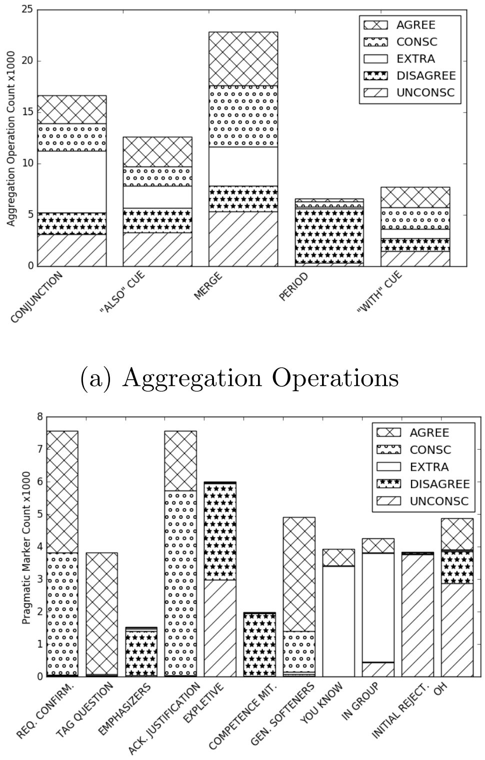
personality given each unique MR, we want to generate a reasonably sized dataset that we can use to train NNLG models, and that exhibits a uniform distribution of personalities for our experiments.

We also point to the fact that the statistical nature of PERSONAGE means that while data generation errors are extremely rare (none found in a manual inspection of a subset of the training data), there is still a chance of error within the training data. We accept this potential error as a consequence of data generation, pointing to the fact that even crowd-sourcing leads to some noise within the E2E data itself: Juraska et al. [Juraska and Walker, 2018] report an 8.48% slot error rate (e.g. missing or repeated values from the MR appearing in the reference) within test references in E2E.

Data Split	**# Unique MRs**	**# Refs per Personality**	**Total**
Train+Dev	3,784	17,771	88,855
Test	278	1	1,390
Total	4,062	17,772	90,245

Table 4.4: Data split in the PersonageNLG corpus [Oraby et al., 2018b].

Figure 4.3 shows the distribution of different aggregation operations and pragmatic markers in the PERSONAGENLG training corpus. From the figure, we see that while each personality type distribution can be characterized by a single stylistic label (the personality), in reality each distribution is characterized by multiple interacting stylistic parameters. It is critical to note that different personalities have *overlapping* style parameters, i.e. particular markers show up across multiple personalities, as can be seen from the figure. For example, in terms of aggregation operations, while several personalities are characterized by the use of aggregation in the form of “conjunction”, the EXTRAVERT personality uses conjunction most frequently.



(a) Aggregation Operations

(b) Pragmatic Markers

Figure 4.3: Frequency of aggregation and pragmatic markers in PersonageNLG train.

The strength of the PERSONAGENLG corpus is that it provides a large set of data that is grounded in well-defined semantic and stylistic properties, which are known in advance for every instance. In the next section, we move on to describe how we use the PERSONAGENLG corpus to explore how a neural model may be trained to disentangle style from content, and faithfully produce semantically correct utterances that vary style.

4.3 Neural Models for Producing Style

We base our seq2seq models for exploring stylistic control in NNLG around the open-source baseline model in the E2E challenge, TGen[2] [Dušek and Jurcicek, 2015, Dusek and Jurcícek, 2016], which we described in detail in Chapter 3. In this section, we describe how we use this underlying framework to train models with increasing levels of stylistic supervision.

4.3.1 Architecture Overview

To recap, TGen is a seq2seq encoder-decoder generation framework with attention [Bahdanau et al., 2014, Sutskever et al., 2014], and is implemented in Tensorflow [Abadi et al., 2016]. The model uses a sequence of Long Short-Term Memory (LSTM) cells [Hochreiter and Schmidhuber, 1997] for the encoder and decoder, a multiplicative attention unit, and a combination of beam-search and reranking for output tuning.

As we described in Section 3.3, the input to the E2E baseline model is a sequence of entries representing the input MR for each training instance. Specifically, each entry is actually a concatenation of three tokens: a dialog act (e.g. *inform*), an attribute (e.g. *food*), and a value (e.g. *Chinese*). To preprocess our training examples from the PersonageNLG corpus, we first delexicalize all MR-NL pairs that include instances of attributes that take on proper-noun values, i.e. *name* and *near*, in order to prevent excessive data sparsity (they are relexicalized in a post-processing phase at generation time). The resultant sequence of entries is sorted by dialog act and attribute name, and then each token is internally represented as an

[2] TGen source code: `https://github.com/UFAL-DSG/tgen`

embedding of floating-point numbers, which are randomly initialized and updated during training.

The general model architecture we use is summarized in Figure 4.4.[3] We use the same underlying framework for all of our experiments, but train different models with increasing amounts of stylistic supervision, in order to systematically test how well they are able to balance semantic fidelity and stylistic variation. We describe each model in the following sections, referring back to the specific architectural differences in our discussion.

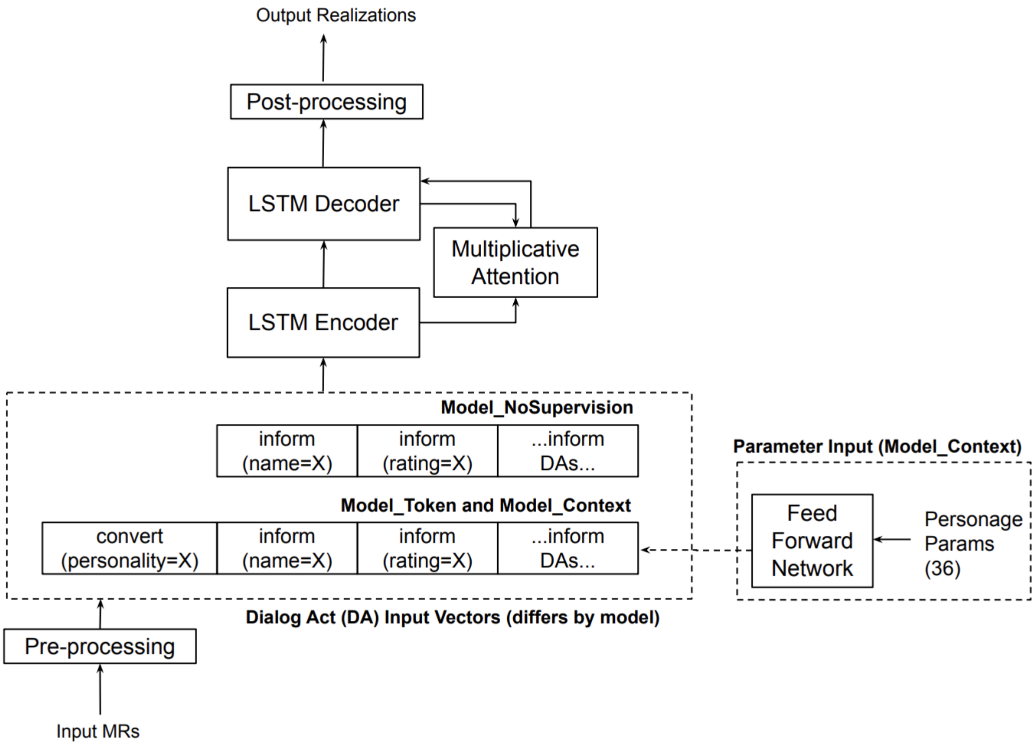


Figure 4.4: General architecture for neural style models [Oraby et al., 2018b].

[3] We showed a more detailed model diagram of a standard seq2seq architecture in Figure 3.2, but here, we focus on summarizing the architectural differences between our style models.

4.3.2 Style Encoding for Model Control

We develop three different models to systematically test the effects of increasing the level of supervision, with novel architectural additions to accommodate these changes. Our baseline model, MODEL_NOSUPERVISION, is the default baseline model from the E2E challenge. Our second model, MODEL_TOKEN, uses a single additional token in the input MR to define the personality of the corresponding NL. The most supervised model, MODEL_CONTEXT, encodes a detailed set of PERSONAGE parameters directly into the model along with the input MR. We describe each model in more detail below.

Model_NoSupervision

The simplest model follows the baseline TGEN architecture [Dusek and Jurcícek, 2016], where each PERSONAGENLG MR is paired with its corresponding NL, *without any personality information in the MR.* Effectively, what this means is that each input MR in training is *repeated* five times, paired once with each of the five personality-specific realizations. This setting serves as a baseline for our experiments, showing how the model will perform without any specific instructions on the different personality styles.

Model_Token

MODEL_TOKEN follows the same setup as MODEL_NOSUPERVISION, but this time includes a single added supervision token indicating the personality type for each MR. This extra token is shown in the dialog act input box at the bottom of Figure 4.4, where in addition to the sequence of entries (where each is a concatenated triple of dialog act, attribute, and value), there is an additional entry, *convert(personality=X)*. Specifically, this extra token is structured as a concatenated triple as with every other MR entry, but specifies a new dialog act, *convert*, which defines a *personality* attribute, whose value is the PERSONAGE personality of the corresponding MR-NL instance.

This model is inspired by the use of a language token for machine translation [Johnson et al., 2016]. In the case of neural machine translation, the token is included at the beginning of the input sequence to dictate the required target language for translation. Unlike other work that uses a single token to control generator output [Fan et al., 2017, Hu et al., 2017], this personality token encodes a constellation of different parameters that define the style of the matching reference, specifically, the array of aggregation and pragmatic marker operators discussed in Section 4.2. Uniquely here, the model attempts to *simultaneously* control multiple style variables that may interact in different ways.

Model_Context

While MODEL_TOKEN provides a high-level abstraction of the combination of style choices that jointly function to express an MR's content in a particular personality style in the PERSONAGENLG corpus, the specific style choices that go into the NL are underspecified. In PERSONAGE itself, aggregation and pragmatic marker choices are internally defined as operations on syntax trees based on the attributes in an MR: since no such syntactic structure is provided at any stage to the end-to-end neural model, a model must thus derive latent representations that function *as though* they also operate on syntactic trees in order to mimic these operations correctly.

In our most supervised model, Model_Context, we provide the neural model with *some* guidance as to the types of syntactic and stylistic choices to make for a given MR-NL pair, not by providing it with the specific syntactic trees that are required as input to a statistical model like Personage, but instead by providing it with a set of boolean features identifying the existence/non-existence of specific stylistic choices. Specifically, we define a set of 36 style features based on the aggregation operations and pragmatic markers defined in Tables 4.2 and 4.3, each specifying whether or not that specific style feature appears in the NL. Since we used these parameters within Personage to generate the NL for each instance, we have full knowledge of which parameters occur in each instance.

Table 4.5 shows the list of 36 parameters we use to summarize the style features for each MR to NL instance, with examples of the active parameters in two data instances in Table 4.6. In the case of **MR 1** in Table 4.6, for example, all 36 style parameters would be set to False, except the 3 active parameters specified in the last column.

1: emph-you-know
2: emph-really
3: emph-basically
4: emph-actually
5: emph-just
6: emph-exclaim
7: down-kind-of
8: down-sort-of
9: down-somewhat
10: down-quite
11: down-rather
12: down-around
13: down-like
14: down-err
15: down-mmhm
16: down-subord
17: down-i-mean
18: ack-yeah
19: ack-well
20: ack-oh
21: ack-right
22: ack-ok
23: ack-i-see
24: expletives
25: near-expletives
26: comp-mitigation
27: in-group-marker
28: init-rejection
29: tag-question
30: req-confirm
31: aggreg-conjunct
32: aggreg-ellipses
33: aggreg-also
34: aggreg-merge
35: aggreg-period
36: aggreg-with

Table 4.5: The 36 parameters used to encode style in Model_Context.

To encode this additional style information in Model_Context, we fol-

MR 1: personality[disagreeable], name[<name>], food[Japanese], customerRating[low], familyFriendly[yes], near[<near>] **Ref 1:** *<name> is a Japanese restaurant,* **also** *it is kid friendly***.** *It has* **like***, a low rating. It is near <near>.*	13: down-like 33: aggreg-also 35: aggreg-period
MR 2: personality[conscientious], name[<name>], food[Chinese], customerRating[average], area[riverside], familyFriendly[no] **Ref 2: Let's see**, *<name>...* **I see, well** *it isn't kid friendly,* **also** *it's a Chinese restaurant* **sort of** *in riverside,* **also** *it has an average rating.*	8: down-sort-of 19: ack-well 23: ack-i-see 30: req-confirm 33: aggreg-also 34: aggreg-merge

Table 4.6: Sample MR-to-NL pairs and the corresponding active style parameters from Table 4.5.

low a similar approach to Dusek et al. [Dusek and Jurcícek, 2016], who show that encoding additional information about previous turns in a dialog allows for more context-aware responses from a neural-based spoken dialog system (also trained using MR to NL pairs). In their case, they experiment with different ways of encoding the string text of the previous dialog turns, such as prepending it to the input MR, and encoding it separately before concatenating it with the encoded MR, and find that incorporating the context improves their performance on the response generation task (through both quantitative and qualitative evaluations).

Specifically in our case, we encode the 36 style parameters as a *context vector*, as shown at the bottom right of Figure 4.4. The parameters for each reference text are encoded as a boolean vector, and a feed-forward network is used as a context encoder, concatenating the vector onto the MR as input to the hidden state of the encoder and the multiplicative attention unit. The activations of the fully connected nodes are represented as an additional time step of the encoder of the seq2seq architecture [Sutskever et al., 2014]. The attention [Bahdanau et al., 2014] is computed over all of the encoder states (i.e. both the input MR and additionally encoded context vector), and the hidden state of the fully connected network. Other work has also explored context representations of the prior dialog for response generation. Sordoni et al. [Sordoni et al., 2015] incorporate previous utterances as a bag of words model and use a feed-forward neural network to inject a fixed sized context vector into the LSTM cell of the encoder. Ghosh et al. [Ghosh et al., 2016] proposed a modified LSTM cell with an additional gate that incorporates the previous context as input during encoding. The weights of the gate are learned exactly in the same way the weights for the input, forget, and output gates. We use a similar context representation to encode stylistic parameters for our NNLG models, but our input is not contextual in the sense of an ongoing set of turns in a dialog, but instead encodes a contextual representation of parameters defining the composition of our NL realization.

4.3.3 Model Configurations

We train all of our models using the 88k training MR to NL pairs from the PersonageNLG corpus, reserving 2k instances for parameter tuning. At test time, we generate a single output per test MR-NL pair, thus resulting in 1,390 test outputs for each of our three models (that can be compared with the 1,390 Personage references from the test set).

For batch size and beam size, we use the default parameter setting from TGen (20 and 10, respectively). We use an embedding size of 50 for input encoding, and monitor loss on the validation set to set the number of epochs (ultimately setting a maximum of 20 epochs with a minimum of 5 passes, and early stopping). Table 4.7 summarizes the precise parameters we use for our most complex model,

Model_Context. All experiments are run on an Nvidia Tesla K20c GPU with the GPU version of TensorFlow.

validation-size: 2000	alpha: 5e-4
epochs: 20	max-cores:4
min-epochs: 5	max-mr-len: 11
improve-interval: 5	hidden-units: 128
beam-size: 10	embed-size: 50
context-len: 36	batch-size: 20
alpha-decay: 3e-2	optimizer: adam

Table 4.7: Summary of model configurations for Model_Context.

4.4 Evaluating Semantics and Style

As we described in detail in Sections 2.5 and 3.4, it is widely agreed that new evaluation metrics are needed for NLG [Langkilde-Geary, 2002, Belz and Reiter, 2006, Bangalore et al., 2000, Novikova et al., 2017a]. Specifically in our case, we are interested in evaluating *both* how well our models do at the standard task of semantic preservation, where some metrics already exist, but also on the novel task of style generation, particularly on whether our models are able to produce the required aggregation and pragmatic marker operations. Thus, we structure our model evaluations here into quantitative and qualitative metrics, each targeting a specific semantic or stylistic characterization to compare our models.

Table 4.8 shows examples of the output of all three models, given different input MRs. We note, of course, that each model is provided with slightly different inputs: Model_NoSupervision is not provided the personality token in the MR, Model_Token *is* provided the personality token, and Model_Context is provided both the personality token in the MR *and* the context vector of 36 style parameters. We refer to these model outputs in our discussion of different metrics below.

MR 1: personality[unconscientious], name[<name>], near[<near>], area[city centre], eatType[restaurant], familyFriendly[no], food[fast food]

Model_NoSupervision: *Let 's see what we can find on <name>. I see, well it is a fast food place near <near> in city centre and it isn 't kid friendly. It is a restaurant.*

Model_Token: *Oh god yeah, i am not sure. <name> is a fast food place in city centre, also it is a fast food place, also it is n 't kid friendly, also it 's a restaurant.*

Model_Context: *Oh god yeah, i don't know. Mmhm... <name> is a fast food place near <near> in city centre and it is a restaurant, also it isn 't family friendly.*

MR 2: personality[extravert], name[<name>], near[<near>], area[riverside], eatType[restaurant], familyFriendly[yes], food[English], priceRange[20-25]

Model_NoSupervision: *Let 's see what we can find on <name>. Well, i see it is an english place near <near> with a price range of 20-25, also it is kid friendly, also it 's a restaurant, also it is in riverside.*

Model_Token: *<name> has a price range of 20-25 with a price range of 20-25, you know and it is family friendly near <near> and an english place, you know!*

Model_Context: *<name> is a restaurant, it has a price range of 20-25, you know, it is near <near> and it is kid friendly, also it 's an english place, also it is in riverside, you know!*

Table 4.8: Sample model outputs for all three style models given an input MR.

4.4.1 Quantitative Evaluation

We divide our quantitative evaluations in two parts: first looking at different metrics for evaluating semantic quality, and then at metrics for evaluating style control. For semantic quality, we first present the standard set of automated metrics used in NLG to measure how well model outputs conform to the Personage test references. Then, we present a more detailed exploration of the specific *types* of semantic mistakes each model makes (i.e. deletions, repetitions, and substitutions). In terms of stylistic variation, we first evaluate the entropy for the set of outputs for each model, then we explore how well the outputs of each model correlate with the aggregation and pragmatic marker operations.

Evaluating Semantic Quality

Automatic Metrics. We begin our evaluation with the standard automatic metrics used for evaluating translation/generation system outputs, compared to "gold-standard" references. We use the evaluation scripts provided with the E2E generation challenge[4] The evaluation scripts work by taking in a set of model outputs and a set of references (in our case, the PERSONAGE-generated references for the test set from our PERSONAGENLG corpus), and computing the average score for each metric for all pairs of NL output-to-reference.

Table 4.9 summarizes the results for the metrics used, specifically BLEU (n-gram precision), NIST (weighted n-gram precision), METEOR (n-grams with synonym recall), and ROUGE (n-gram recall), and CIDEr (TF-IDF weighted n-gram cosine similarity), as described in Section 2.5.1. Although the differences are small, MODEL_CONTEXT, with the extra input parameters and context embedding, shows the highest averages across all of the metrics. We also note that MODEL_TOKEN scores better than MODEL_NOSUPERVISION for all metrics.[5]

Model	**BLEU**	**NIST**	**METEOR**	**ROUGE_L**	**CIDEr**
NOSUPERVISION	0.2774	4.2859	0.3488	0.4567	1.3096
TOKEN	0.3464	4.9285	0.3648	0.5016	1.6886
CONTEXT	**0.3766**	**5.3437**	**0.3964**	**0.5255**	**1.9380**

Table 4.9: Automatic metric evaluations.

[4] Eval scripts: https://github.com/tuetschek/e2e-metrics

[5] There have been subsequent experiments in collaborative work using different architectures, achieving even higher automatic metric scores [Harrison et al., 2019].

Deletions, Repetitions, and Substitutions.

Automated evaluation metrics are not informative about the quality of the outputs, and penalize models for introducing stylistic variation [Novikova et al., 2017a]. We thus develop scripts to automatically evaluate the common types of neural generation errors by comparing the input MR to the output model NL: *deletions* (failing to realize a value), *repeats* (repeating a value), and *substitutions* (mentioning an attribute with an incorrect value).

Table 4.10 shows examples of each mistake type: the deletion example shows how the price range attribute is missing from the realization; the repetition example shows repeated mentions of the family friendly and food attributes; and the substitution example shows how the eat-type attribute is falsely realized with value "coffee shop" instead of "restaurant".

Table 4.11 shows ratios for the number of deletions, repetitions, and substitutions for each model for the test set. The error counts are split by personality, and normalized by the number of unique MRs in the test set (278). Smaller ratios are preferable, indicating fewer errors. Note that because MODEL_NOSUPERVISION does not encode a personality parameter, the error values are the same across each personality (averages across the full test set).

The table shows that MODEL_NOSUPERVISION makes very few semantic errors (we show later that this is at the cost of limited stylistic variation). Across all error types, MODEL_CONTEXT makes significantly fewer errors than MODEL_TOKEN, suggesting that its additional explicit parameters help avoid semantic errors. The last row quantifies whether some personalities are harder to model: it shows that

1. Deletion: priceRange[high]

MR: personality[disagreeable], name[<name>], near[<near>], eatType[restaurant], area[city centre], **priceRange[high]**, familyFriendly[no], food[fast food]

Output: *<name> is n 't kid friendly, also it 's a fast food place. It is n 't kid friendly. It is a fast food place in city centre.*

2. Repetition: food[Italian], familyFriendly[no]

MR: personality[disagreeable], name[<name>], near[<near>], eatType[pub], area[city centre], priceRange[high], **familyFriendly[no], food[Italian]**

Output: *<name>* **is n 't kid friendly**, *also it 's an* **italian place**. *It 's in city centre. It* **is n 't family friendly**. *It is an* **italian place**. *It is a pub.*

3. Substitution: eatType[restaurant] → eatType[coffee shop]

MR: personality[disagreeable], name[<name>], near[<near>], **eatType[restaurant]**, area[riverside], priceRange[moderate], familyFriendly[yes], food[English], customerRating[decent]

Output: *<name> has like, a decent rating, also it is damn moderately priced. It is a* **coffee shop**. *It is near <near>. It is an english place. It is damn kid friendly.*

Table 4.10: Examples of deletion, repetition, and substitution mistakes in model output.

across all models, DISAGREEABLE and EXTRAVERT have the most errors, while CONSCIENTIOUS has the fewest.

Model	AGREE	CONSC	DISAG	EXTRA	UNCON
DELETIONS					
NOSUPERVISION	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
TOKEN	0.27	0.22	0.87	0.74	0.31
CONTEXT	0.08	**0.01**	0.14	0.08	**0.01**
REPETITIONS					
NOSUPERVISION	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
TOKEN	0.29	0.12	0.81	0.46	0.28
CONTEXT	0.02	**0.00**	0.14	**0.00**	**0.00**
SUBSTITUTIONS					
NOSUPERVISION	0.10	0.10	0.10	0.10	0.10
TOKEN	0.34	0.41	0.22	0.35	0.29
CONTEXT	**0.03**	**0.03**	**0.00**	**0.00**	**0.03**
All	0.68	0.35	1.96	1.29	0.61

Table 4.11: Ratio of model errors by personality.

We point back to the model output examples from Table 4.8: in both examples, both Model_NoSupervision and Model_Context make no deletions, repetitions, or substitution errors. Model_Token, on the other hand, repeats the value for "family friendly" in the first realization, and for "price range" in the second. It also deletes the value for "area" in the second example.

Evaluating Stylistic Variation

Entropy. As a general measure of how varied our model outputs are, we measure Shannon text entropy, to quantify the amount of variation in the output produced by each model as a function of the different n-gram sequences produced by each model in aggregate. We calculate entropy as $-\sum_{x \in S} \frac{freq}{total} * log_2(\frac{freq}{total})$, where S is the set of unique words in all outputs generated by the model, $freq$ is the frequency of a term, and $total$ counts the number of terms in all references. We compute entropy for the original set of references from the PersonageNLG test set as a baseline, and then in turn for each set of model outputs.

Table 4.12 shows the entropy values we find for each set, at different levels of n-grams. Naturally, the input training data has the highest entropy, but we observe that Model_Context has the highest entropy of all three model outputs. This implies that it is the best at preserving the variation seen in the training data.

This finding is most exciting when considering the findings regarding semantic errors from Table 4.11: we saw that Model_NoSupervision makes the fewest semantic errors, but here we see that it produces the least varied output. Model_Token does produce output that is comparably varied as measured by entropy, but does the most poorly on semantic errors. Model_Context, informed by the explicit stylistic context encoding, makes comparably few semantic errors, while producing stylistically varied output with high entropy, nearing that of the test references.

Referring back to the model outputs in Table 4.8, we see a clear example of the entropy bottleneck for MODEL_NOSUPERVISION. Although the two input MRs are distinct and the personalities are different, MODEL_NOSUPERVISION returns an NL realization that is almost identical in structure for both MRs. The realizations for MODEL_TOKEN and MODEL_CONTEXT are notably different for both MRs, and it is notable that both models make many of the same stylistic choices, e.g. *initial rejection* ("I don't know") for the UNCONSCIENTIOUS example, and *expletives* ("Oh God") in the DISAGREEABLE example (although as we saw earlier, MODEL_TOKEN does this at the cost of accurate semantics).

Model	**1-grams**	**1-2grams**	**1-3grams**
PERSONAGETRAIN	5.97	7.95	9.34
NOSUPERVISION	5.38	6.90	7.87
TOKEN	5.67	7.35	8.47
CONTEXT	**5.70**	**7.42**	**8.58**

Table 4.12: Shannon Text Entropy

Aggregation. To measure the ability of each model to aggregate, we compare the model's aggregation behavior to that of the test references. We divide each model output by personality, then for each instance, we compute a vector of booleans defining whether or not the model produced each possible aggregation operation (we identify these automatically by searching in the NL). We do the same for the test references, and compute a Pearson correlation between the full set of boolean vectors for a given model output and personality, as compared to the respective set of test outputs. Table 4.13 shows these correlations for each model and personality.

Model	**AGREE**	**CONSC**	**DISAG**	**EXTRA**	**UNCON**
NoSupervision	0.78	0.80	0.13	0.42	0.69
Token	0.74	0.74	**0.57**	0.56	0.60
Context	**0.83**	**0.83** .	0.55	**0.66**	**0.70**

Table 4.13: Correlations between Personage and models for aggregation operations in Table 4.2

The correlations in Table 4.13 (all significant with $p \leq 0.001$ compared to the references) show that Model_Context has a higher correlation with Personage than the two simpler models (except for disagreeable, where Model_Token is higher by 0.02). Here, Model_NoSupervision actually *frequently* outperforms the more informed Model_Token. Note that *all personalities use aggregation*, even though **not** *all personalities use pragmatic markers*, and so even without a special *personality* token, we see that Model_NoSupervision is able to faithfully reproduce aggregation operations common within the training data. In fact, since the correlations are frequently higher than those for Model_Token, we hypothesize that it is able to more accurately focus on aggregation (common to all personalities) than stylistic differences, which Model_Token is able to produce with the guidance of the single personality token.

Pragmatic Marker Usage. To measure whether the trained models faithfully reproduce the pragmatic markers for each personality, we again compute Pearson correlation between the personage references and the outputs for each model and personality for all pragmatic markers. Table 4.14 show the results of this analysis (all correlations significant with $p \leq 0.001$ compared to the references).

From Table 4.14, we again see that Model_Context has the highest correlation with the training data, for all personalities (except agreeable, with significant margins, and conscientious, with a margin of just 0.01). It is interesting to note that Model_NoSupervision shows positive correlation with agreeable and conscientious, it shows *negative* correlation with the Personage inputs for disagreeable, extravert, and unconscientious. The pragmatic marker distributions for Personage train in Figure 4.3 indicate that the conscientious personality most frequently uses *acknowledgement-justify* (i.e., *"well"*, *"i see"*), and *request confirmation* (i.e., *"did you say X?"*), which are less complex to introduce into a realization since they often lie at the beginning or end of a sentence, allowing the simple Model_NoSupervision to learn them.[6]

We reiterate here that the personalities have overlapping style parameters, such that our finite set of aggregation and pragmatic marker operators show up across multiple personalities (with different distributions, as demonstrated in Figure 4.3). Since Model_Context is provided with the most explicit information about the aggregation and pragmatic marker choices characterizing a given training input, we hypothesize this is why it is better suited to discriminate between the fine-grained differences between personalities as compared to the other models. We note that if there was no overlap between the operations each personality produces, we might expect that simple Model_Token supervision may have been enough to accurately reproduce each personality's style choices at test time.

[6] We verified that there is not a high correlation between every set of pragmatic markers: different personalities do not correlate, e.g., -0.078 for Personage disagreeable and Model_Token agreeable.

Model	AGREE	CONSC	DISAG	EXTRA	UNCON
NoSupervision	0.05	0.59	-0.07	-0.06	-0.11
Token	**0.35**	0.66	0.31	0.57	0.53
Context	0.28	**0.67**	**0.40**	**0.76**	**0.63**

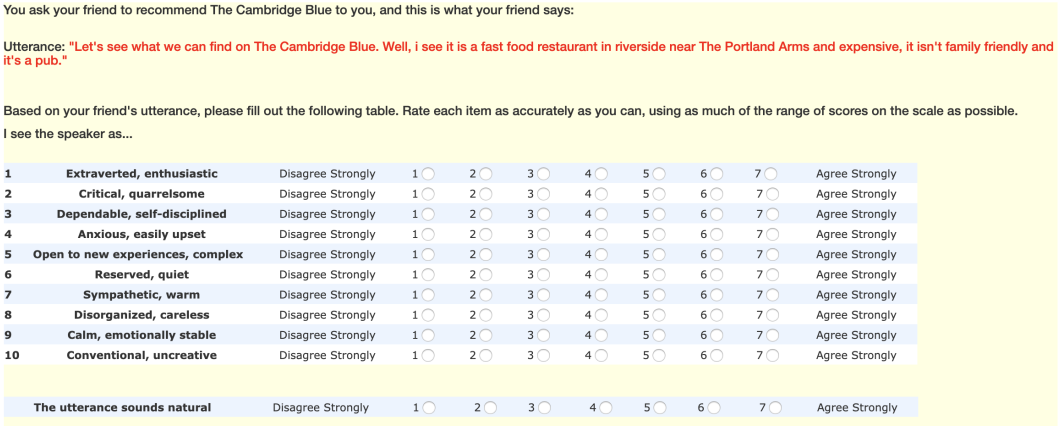
Table 4.14: Correlations between Personage and models for pragmatic markers in Table 4.3

4.4.2 Qualitative Analysis

Crowdsourcing Personality Judgments. Based on our quantitative results, we select Model_Context as the best-performing model overall, and conduct an evaluation to test if humans can distinguish the personalities exhibited. We randomly select a set of 10 unique MRs from the Personage training data along with their corresponding reference texts for each personality (50 items in total), and 30 unique MRs Model_Context outputs (150 items in total).

We construct a HIT on Mechanical Turk, as shown in Figure 4.5, presenting a single output (either Personage or Model_Context), and ask 5 Turkers to label the output using the Ten Item Personality Inventory (TIPI) [Gosling et al., 2003]. The TIPI is a ten-item measure of the Big Five personality dimensions, consisting of two items for each of the five dimensions, one that *matches* the dimension, and one that is the *reverse* of it, and a scale that ranges from 1 (disagree strongly) to 7 (agree strongly). To qualify Turkers for the task, we ask that they first complete a TIPI on themselves, to help ensure that they understand it.

Table 4.15 presents results as aggregated counts for the number of times at least 3 out of the 5 Turkers rated the *matching* item for that personality higher than the *reverse* item (Ratio Correct), the average rating the correct item received (range



You ask your friend to recommend The Cambridge Blue to you, and this is what your friend says:

Utterance: "Let's see what we can find on The Cambridge Blue. Well, i see it is a fast food restaurant in riverside near The Portland Arms and expensive, it isn't family friendly and it's a pub."

Based on your friend's utterance, please fill out the following table. Rate each item as accurately as you can, using as much of the range of scores on the scale as possible.

I see the speaker as...

1	**Extraverted, enthusiastic**	Disagree Strongly	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	Agree Strongly
2	**Critical, quarrelsome**	Disagree Strongly	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	Agree Strongly
3	**Dependable, self-disciplined**	Disagree Strongly	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	Agree Strongly
4	**Anxious, easily upset**	Disagree Strongly	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	Agree Strongly
5	**Open to new experiences, complex**	Disagree Strongly	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	Agree Strongly
6	**Reserved, quiet**	Disagree Strongly	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	Agree Strongly
7	**Sympathetic, warm**	Disagree Strongly	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	Agree Strongly
8	**Disorganized, careless**	Disagree Strongly	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	Agree Strongly
9	**Calm, emotionally stable**	Disagree Strongly	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	Agree Strongly
10	**Conventional, uncreative**	Disagree Strongly	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	Agree Strongly

The utterance sounds natural	Disagree Strongly	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	Agree Strongly

Figure 4.5: HIT interface for TIPI judgments.

between 1-7), and an average "naturalness" score for the output (also rated 1-7). From the table, we can see that for PERSONAGE training data, all of the personalities have a correct ratio that is higher than 0.5. The MODEL_CONTEXT outputs exhibit the same trend except for UNCONSCIENTIOUS and AGREEABLE, where the correct ratio is only 0.17 and 0.50, respectively (they also have the lowest correct ratio for the original PERSONAGE data). It is interesting to note that the difficulty annotators have in distinguishing the UNCONSCIENTIOUS personality dimension in particular lines up with findings in previous work on the perception of PERSONAGE output [Mairesse and Walker, 2011].

Table 4.15 also presents results for naturalness for both the reference and generated utterances, showing that both achieve decent scores for naturalness (on a scale of 1-7), all above 4.3 for MODEL_CONTEXT and above 4.2 for PERSONAGE. While human utterances would likely be judged more natural, it is not at all clear that similar experiments could be done with human generated utterances, where it is difficult to enforce the same amount of experimental control.

	Personage			Model_Context		
Person.	**Ratio Correct**	**Avg. Rating**	**Nat. Rating**	**Ratio Correct**	**Avg. Rating**	**Nat. Rating**
AGREE	0.60	4.04	5.22	0.50	4.04	4.69
DISAGR	0.80	4.76	4.24	0.63	4.03	4.39
CONSC	1.00	5.08	5.60	0.97	5.19	5.18
UNCON	0.70	4.34	4.36	0.17	3.31	4.58
EXTRA	0.90	5.34	5.22	0.80	4.76	4.61

Table 4.15: Percentage of correct items and average ratings and naturalness scores for each personality (Personage vs. Model_Context).

4.5 Generalizing to Multiple Personalities

Thus far, we showed that we can augment the E2E training data with synthetically generated stylistic variants and train a neural generator to reproduce these variants; however, even the best-performing Model_Context can naturally only generate what it has seen in training [Oraby et al., 2018b]. Here, instead, we explore whether a model that is trained to achieve a single stylistic personality target can produce outputs that combine stylistic targets, to yield a novel style that is significantly different from what was seen in training, while still maintaining high semantic correctness.

For this experiment, we train a new model, Model_Multivoice, which is similar to Model_Token from our previous experiments, providing it with training instances with single personality tokens as before. Then, at generation time, we provide the model with test instances that contain *two* personality tokens: something it has never seen in training, i.e. we instruct the model to generate "multivoice" outputs that combine two personalities, for example extravert with disagreeable, where such combined outputs never occurred in the training data [Oraby et al., 2018a].

We note that this new model differs from the Model_Token we used earlier because it is trained on *unsorted inputs* to allow us to add multiple personality tags to the MR at generation time, and to differentiate between test MRs that ask for *personality[extravert], personality[agreeable]* from asking for *personality[agreeable], personality[extravert]*, since these may be modeled differently (more weight to one personality's style choices). Note that we do not train on multiple personalities, instead, we train one model that uses all the data, where each distinct single personality has a corresponding personality token in the training instance, and each test instance has two. Also note that we use the simpler method of including a personality token as in Model_Token, instead of using the context vector method from Model_Context, since it is unclear how to cleanly generate a combined vector for the two-personality examples.

To generate our test set for the multivoice experiments, we generate 2 references per combination of two personalities for each of the 278 test MRs, since the order of the convert tags matters. For a given order, the model produces a single output. We do not combine personalities that are exact opposites such as agreeable and disagreeable, yielding 8 combinations. Thus, the multivoice test set consists of 4,448 total realizations (278 MRs and 8 × 2 outputs per MR).

Sample outputs are given in Table 4.16 for the disagreeable personality, which is one of the most distinct in terms of aggregation and pragmatic marker insertion, as it is combined in the multivoice setting with Conscientious (Row 3), Extravert (Row 5), and Unconscientious (Row 7). We also show single personality outputs for each respective personality (Rows 2, 4, and 7) for comparison.

To quantify how Model_Multivoice learns new combinations of the aggregation and pragmatic marker choices when tested on a pair of personalities, we compute occurrence counts (frequency shown scaled down by 100) for the *period aggregation* and *expletive pragmatic markers*, which are the two operations DISAGREEABLE performs most frequently. For the single personalities, we compute the frequencies from training instances, and for the multivoice setting, we compute them from the generated test outputs.

We can see from the table that while CONSCIENTIOUS on its own realizes the content in two sentences, period aggregation is much more prevalent in the DISAGREEABLE + CONSCIENTIOUS example, with the same content being realized in 5 sentences. Also, we see that some of the expletives originally in DISAGREEABLE are dropped in the combined output. The same is true in the case of DISAGREEABLE + EXTRAVERT and DISAGREEABLE + UNCONSCIENTIOUS: period aggregation is scaled up significantly as compared to the single personality settings. This suggests that MODEL_MULTIVOICE learns a combined representation unlike what it has seen in train, and is able to interpolate between personality parameters.

4.6 Summary

In this chapter, our goal was to evaluate the ability of different neural architectures to faithfully render semantic content, defined by an input MR, while simultaneously exhibiting stylistic variations. To set up an environment where we have control over both semantics and style, we created a novel parallel training corpus, PERSONAGENLG, that builds on the E2E generation challenge corpus. It contains over 88,000 meaning representations in the restaurant domain, and matched reference outputs that express the input content in five different Big-Five personalities,

MR: inform(name[Browns Cambridge], eatType[pub], priceRange[average], food[Italian], near[Adriatic], familyFriendly[yes], area[city centre])

	Persona	**Period Agg.**	**Explet. Prag.**	**Example**
1	DISAG	5.71	2.26	Browns Cambridge is damn moderately priced, also it's in city centre. It is a pub. It is an Italian place. It is near Adriatic. It is damn family friendly.
2	CONSC	0.60	0.02	Let's see what we can find on Browns Cambridge. I see, well it is a pub, also it is moderately priced, an Italian restaurant near Adriatic and family friendly in city centre.
3	DISAG+ CONSC	3.81	0.84	Browns Cambridge is an Italian place and moderately priced. It is near Adriatic. It is kid friendly. It is a pub. It is in city centre.
4	EXTRA	0.18	0.01	Browns Cambridge is a pub, also it is moderately priced, an Italian place near Adriatic, also it is family friendly, you know and it's in city centre, you know!
5	DISAG+ EXTRA	8.07	1.92	Damn moderately priced Browns Cambridge is it 's a pub, also it is near Adriatic. It is an Italian place and moderately priced. It is in city centre.
6	UNCON	0.40	2.88	Oh god yeah, i don't know. Browns Cambridge is a pub, also it is damn family friendly, also it's an Italian place near Adriatic, also it is darn moderately priced in city centre.
7	DISAG+ UNCON	2.88	3.16	Oh god i mean, i thought everybody knew that Browns Cambridge is a pub, also it is near Adriatic. It is an Italian place and moderately priced. It is in city centre.

Table 4.16: A comparison of single and multivoice generation outputs for Disagreeable, Extravert and Conscientious for a given MR.

generated by using an existing statistical natural language generator, Personage [Mairesse and Walker, 2010]. Personage allows us to take in E2E MRs that define semantics, and systematically generate data that exhibits particular predefined aggregation and pragmatic marker operations that are clearly marked, thus defining style choices. We described the style choices from the Personage generator, as well as our corpus creation method, in Section 4.2.

Given the PersonageNLG corpus, we then moved on to design three neural models that systematically encode more stylistic information into the network in Section 4.3. Our first model, Model_NoSupervision, does not include any style information. Model_Token includes a single personality token identifying the personality of the given MR and NL pair, and finally Model_Context includes a context vector defining 36 unique style parameters describing the aggregation and pragmatic marker operations present in the NL.

In Section 4.4, we conduct a series of rigorous semantic and stylistic evaluations, we find that while Model_NoSupervision is comparably the best at semantic preservation, it is unable to produce any variable style choices. Model_Token does a better job at replicating the style choices given a personality token, but does poorly in terms of semantic fidelity. Only Model_Context, armed with an array of style supervision features, is able to adequately produce required style choices without losing semantic quality. We also verify that the personality styles produced by Model_Context are distinguishable through a human evaluation. In an additional experiment in Section 4.5, we trained a new model, Model_Multivoice, to generate output that combine style choices from two distinct personalities, showing that we can train models to produce output completely novel to what they have seen in training.

While we are able to prove through our experiments that we *can* control style and semantics in a neural generation framework, one major limitation of our methods here is clear: we use synthetically generated data for training our models, and although this method is not uncommon [Weston et al., 2015, Dodge et al., 2015], it means that our outputs are inherently less natural than human-written ones, and that our models are limited in terms of the types of style choices they are exposed to and expected to produce. Thus, in the next chapter, we set out to find more natural data sources for training neural models, with the goal of producing outputs that are as human-like as possible.

Chapter 5

Tackling the Data Bottleneck

“[...] While the world is awash with text waiting to be processed, there are fewer instances of what we might consider appropriate inputs for the process of natural language generation. For researchers in the field, this highlights the fundamental question that always has to be asked: What do we generate from?”

– Introduction to the Special Issue on NLG [Dale et al., 1998]

5.1 Overview

As we have seen, the real power of Neural Natural Language Generation (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 a real **data bottleneck** due to the need for massive amounts of data for training.

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 Meaning Representation (MR), as

1 - E2E [Novikova et al., 2017b] **50k - Crowdsourcing (Domain: Restaurant Description)**
MR: name[Blue Spice], eatType[coffee shop], customer rating[average], near[Burger King] **Human:** *The Blue Spice coffee shop near Burger King has good customer ratings with excellent food and service, with a lovely atmosphere.* **System:** Blue Spice is a pub near Burger King with an average customer rating.
2 - WebNLG [Gardent et al., 2017b] **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 - Laptop [Wen et al., 2016] **13k - Crowdsourcing (Domain: Product Review)**
MR: inform(name=satellite eurus 65; type=laptop; memory=4 gb; driverange=medium; isforbusinesscomputing=false) **Human:** *The satellite eurus 65 is a laptop designed for home use with 4 gb of memory and a medium sized hard drive* **System:** Satellite eurus 65 is a laptop which has a 4 gb memory, is not for business computing, and is in the medium drive range.

Table 5.1: A comparison of popular datasets for NNLG.

we described in Section 2.3. Table 5.1 summarizes three recent efforts: the E2E NLG challenge [Novikova et al., 2017b], the WebNLG challenge [Gardent et al., 2017b], and the Laptop dataset [Wen et al., 2016], all shown with an example of an MR, human reference, and system realization. The largest crowdsourced dataset to date, E2E, consists of 50k instances (which we described in detail in Chapter 3). Other datasets, such as the TV (7k) product review dataset, are similar but smaller [Wen et al., 2015].

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, leaving little room for stylistic variation.

Thus far in this thesis, we have shown how a state-of-the-art Sequence-to-Sequence (seq2seq) model can be trained to produce outputs that do a good job at preserving semantics, and can, with some supervision, be trained to produce outputs that vary stylistically. In our own effort to produce a dataset for stylistic variation in Natural Language Generation (NLG), we created the PersonageNLG corpus, a synthetically-designed dataset of around 88k MR to Natural Language (NL) realizations in the restaurant domain, exhibiting examples of stylistic variation based on Big-Five personalities. Since the data was synthetically generated, it is not as natural as crowdsourced data. But since crowdsourcing requires notable effort and is costly and time-consuming, especially when we need to create large datasets, we wonder whether we can use existing data from the wild to train neural models to learn stylistic choices that are much more similar to how we as humans organically express our communicative goals.

In this chapter, we explore the masses of user review data available online, devising a method to use them to create corpora for training NNLGs. Specifically, we are interested in restaurant reviews from Yelp, where users go to write detailed descriptions of their experiences, frequently using elaborate descriptions and emotionally-charged language. In Section 5.2, we describe a pilot exploration of the types of language and descriptions we observe in such reviews, breaking them down into sentences that describe particular attributes, such as *food* and *staff*.

In Section 5.3, we present our algorithm to create a dataset of MR to NL instances using off-the-shelf tools and open-source resources, without having to do any crowdsourcing. We first describe our method to automatically identify sentences that contain attributes of interest that are semantically-related, and how we systematically mark each sentence with important information to semantically and syntactically characterize it, using review metadata and sentence parse information.

Finally, in Section 5.4, we characterize the YelpNLG corpus of 300k MR to NL sentences, comparing it to existing corpora for NNLG which we have described earlier in this thesis. We show how our corpus is significantly larger, and contains much more diverse language than previously released corpora, making it an exciting dataset for exploring stylistic variation in NNLG.

5.2 Restaurant Reviews as a Source of Style

The restaurant domain has been one of the most common applications for spoken dialog systems for at least 25 years [Polifroni et al., 1992, Whittaker et al., 2002, Stent et al., 2004, Devillers et al., 2004, Gašic et al., 2008]. There has been a tremendous amount of previous work on natural language generation of recommendations and descriptions for restaurants [Howcroft et al., 2013, Wen et al., 2015, Novikova et al., 2016], some of which has even focused on generating stylistically varied restaurant recommendations [Higashinaka et al., 2007b, Mairesse and Walker, 2010, Dethlefs et al., 2014], as we discussed in Section 2.4.

Given this, it is surprising that previous work has not especially noted that restaurant reviews are a fertile source of creative and figurative language. Prominent datasets in the restaurant domain, such as E2E [Novikova et al., 2017b] or SF Restaurants [Wen et al., 2015] are useful for training models to describe restaurant attributes such as *cuisine* or *price*, but miss out on the stylistic variation that is available in descriptions of food, ambiance, or service.

Stars	Review
1/5	*This place is probably the worst thing that ever happened to the history of the known world. [...] The food, however, I initially would want to call unremarkable but I can't. I can't call it unremarkable because it is so incredibly remarkably terrible. [...]*
2/5	*Can't say anything about the food, as we were never served. We never saw a server, even after sitting at our table for 15 minutes. Unacceptable.*
3/5	*I was back here a couple of days ago with my family. And although I remember The food being a lot better than this time around. I was kind of disappointed. The service was okay since I had no Jose this time. Nothing to mention here just refills chips salsa and beverages when you need and food when it's ready.*
4/5	*I would eat here everyday if I didn't think I'd end up 400 pounds... Minus 1 star because each time I've been here the service has kinda sucked and orders have been messed up. Regardless, their fried chicken on waffles topped with syrup and a slice of Red Velvet cake to top it off......... is sooooooo heavenly.*
5/5	*I only have one warning about this restaurant. The food is so amazing that you cannot eat Mexican food anywhere else. [...] I had chicken and beef enchiladas which had homemade corn tortillas and the most tender meat I had ever tasted. [...] I will be a customer for life here!*

Table 5.2: Restaurant reviews by rating from Yelp reviews.

For example, consider the elaborate descriptions in the restaurant reviews in Table 5.2, which come from the Yelp Challenge dataset:[1] e.g. phrases such as *worst thing that ever happened in the history of the known world* along with *incredibly remarkably terrible* (Row 1), *eat here everyday if I didn't think I'd end up 400 pounds* and *sooooooo heavenly* (Row 4), and *food so amazing you cannot eat [...] anywhere else* (Row 5). These phrases express extremely valenced reactions to restaurants, their menu items, and related attributes, using figurative language and interesting descriptions.

The creativity exhibited in these user-generated restaurant reviews can be contrasted with the domains and system output of current NNLG systems, as we

[1] https://www.yelp.com/dataset_challenge

showed earlier in Table 5.1. Rather than simply informing the reader of facts, reviews contain descriptions deeply rooted in experiences, often aiming to be humorous and catchy. Also, it is interesting to note that the reviews contain references to multiple different attributes: for example, Row 5 in Table 5.2 describes that the *"***food** *was amazing"*, going further to say that the restaurant had the *"most tender* **meat***"*. Rows 3 and 4 both mention *service*, e.g. *"the* **service** *was okay"* (Row 3), and *"the* **service** *kinda sucked"* (Row 4).

5.2.1 A Closer Look at Attribute Descriptions in Reviews

Having observed these interesting descriptions about different restaurant domain attributes in user reviews, we conduct a small pilot study to take a closer look at more examples of the types of descriptions we can find within the reviews. To do this on a small scale, we focus on a set of 5 attributes: *restaurant-type, cuisine, food, service,* and *staff*, and construct small lexicons of values that these attributes might take on, inspired by attribute-value pairs from the E2E dataset. For example, *pub* is a valid *restaurant-type*, *Chinese* is a *cuisine*, and *waiter* is a value for the *staff* attribute.

Using these small lexicons, we search for instances of these values in a sample of 20k Yelp reviews, with varied star ratings out of 5 (around 7k positive with 4-5 stars, 5k neutral with 3 stars, and 7k negative with 1-2 stars). We extract sentences that are relatively short, between 5-15 words, and contain any single value of interest, delexicalizing them to allow us to better visualize the syntactic constructions of the sentences these values occur in. Table 5.3 shows examples of delexicalized sentences we extract for both the positive and negative classes. From the table, we see interesting templates emerge: for example, in the positive set, the *cuisine* sentence, *"Wow what a great little <cuisine> joint!"*, the positive *food* sentence, *"The <food> is not cheap, but well worth it."*, or the negative *restaurant* sentence, *"I was appalled by the experience and will not frequent this <restaurant> ever again."*

Attribute	**Template**
Positive	
RESTAURANT	By far my favorite <RESTAURANT> I have ever been to in my life .
CUISINE	Wow what a great little <CUISINE>] joint !
FOOD	The <FOOD> is not cheap , but well worth it.
SERVICE	The <SERVICE> is always friendly and fast .
STAFF	<STAFF> was extremely helpful and knowledgeable and was on top of everything.
Negative	
RESTAURANT	I was appalled by the experience and will not frequent this <RESTAURANT> ever again.
CUISINE	It's your typical <CUISINE> buffet , nothing to rave about .
FOOD	<FOOD> smelled very bad and tasted worse .
SERVICE	We waited another 5 minutes , still no <SERVICE> .
STAFF	I went with 5 friends and our <STAFF> was really rude.

Table 5.3: Examples of learned creative sentence templates by attribute and polarity.

We also take a closer look at the types of descriptions people use for different attributes. We use the AutoSlog-TS shallow parser and weakly-supervised pattern extractor [Riloff, 1996, Riloff and Phillips, 2004] to find adjective patterns frequently associated with different attributes. AutoSlog-TS simply requires two different sets of data, in our case positive and negative reviews, and will use a set of predefined syntactic templates to find lexically-grounded patterns frequent in each class. Because we are particularly interested in descriptive patterns, we specifically use n-gram pattern templates, `AdjAdj`, `AdvAdj`, `AdvAdvAdj`, as in previous work on pattern extraction in different data styles [Oraby et al., 2015, Oraby et al., 2016].

Food Descriptions	
Positive	**Negative**
INSANELY GOOD	ALMOST RAW
SIMPLY PERFECT	VERY FATTY
RIDICULOUSLY GOOD	PREVIOUSLY FROZEN
ALSO INCREDIBLE	COMICALLY BAD
MY FAV	ABSOLUTELY AWFUL
PERFECTLY CRISP	NOT PALATABLE
DEFINITELY UNIQUE	FAIRLY TASTELESS
ALWAYS SO FRESH	PRETTY GENERIC
JUST PHENOMENAL	SO MEDIOCRE
SO DECADENT	SO BLAND
HIGHLY ADDICTIVE	STILL RAW
CONSISTENTLY GREAT	BARELY WARM
WOW AMAZING	PREPACK. FROZEN
PERFECT LITTLE	MOST PATHETIC
EXPERTLY PREPARED	SICKLY SWEET
FRESHLY BAKED	LUKE WARM

Staff Descriptions	
Positive	**Negative**
SUPER HELPFUL	NOT APOLOGETIC
INCREDIBLY FRIENDLY	NOT KNOWLEDGEABLE
SUPER NICE	VERY RUDE
VERY PERSONABLE	TOO BUSY
SO GOOD	FRIENDLY ENOUGH
SO GRACIOUS	JUST HORRIBLE
VERY KNOWLEDGEABLE	NOT ATTENTIVE
SO KIND	VERY PUSH
EXTREMELY PROFESS.	MORE INTERESTED
ALSO FABULOUS	TOO LAZY
EVEN BETTER	EVEN WORSE
STILL AWESOME	EVERY SINGLE
ALWAYS WARM	VERY POOR
ALWAYS ATTENTIVE	SO FEW
ABSOLUTELY BEST	STILL NO
OUR SWEET	VERY UNHAPPY

Table 5.4: Sample of adjective phrase descriptions associated with Food and Staff attributes.

We create sets of (attribute, adjective pattern) based on the relationship between the adjective and the entity ("is", "was", "tasted", etc.). Using this method, we collect 37 restaurant, 30 cuisine, 247 food, 45 service, and 56 staff patterns for positive and 18 restaurant, 9 cuisine, 221 food, 75 service, and 61 staff patterns for negative. Table 5.4 shows example patterns in each class for the food and staff attributes. Again, we observe interesting and strong descriptions: for example, food is frequently described positively as *"insanely good"* or negatively as *"almost raw"*; staff is frequently described positively as *"super helpful"* or negatively as *"not apologetic"* [Oraby et al., 2017].

Our pilot exploration of the sentences and descriptions about simple restaurant attributes in the Yelp challenge corpus suggests that it is worth exploring how to use this kind of data to train NNLGs. We describe our method for converting text-based reviews to structured content for training NNLGs in the next section.

5.3 From Reviews to Structured Content

Our goal is to develop a novel method for creating datasets for NNLG that are based on harvesting and making use of the masses of freely available, highly-descriptive user review data. Since we are primarily concerned with data-to-text generation from a structured MR to an NL, in order to use this data in any meaningful way, we must first find a way to create structured pairs of MRs representing a sentence's meaning, and pair those with corresponding NL sentences.

Rather than starting with an MR and collecting human reference NLs as in previous work on corpus creation for NNLG [Wen et al., 2015, Wen et al., 2016, Novikova et al., 2016, Novikova et al., 2017b, Gardent et al., 2017a], our idea is to begin with the NL references (i.e. the review sentences), and work backwards, systematically constructing MRs for each sentences using dependency parses and rich sets of lexical, syntactic, and sentiment information based on ontological knowledge bases.

Table 5.5 shows an example MR from YelpNLG (to be compared with Table 5.1), consisting of relational tuples of attributes, values, adjectives, and order information, as well as sentence-level information including sentiment, length, and pronouns. We present our method to automatically "retrofit" an MR from an NL using entirely off-the-shelf tools in the following sections.

YelpNLG
300k - Auto. Extraction (Domain: Restaurant Review)

Review Sentence: *The taco was a small flour tortilla topped with marinated grilled beef, asian slaw and a spicy delicious sauce.*

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]

Table 5.5: A sample of the YelpNLG dataset.

5.3.1 Collecting Lexicons of Restaurant Attributes

As in our pilot in Section 5.2, we begin with reviews from the Yelp challenge dataset, 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.[2]

The first step to creating an MR from the review sentences, is to be able to identify a broad range of attributes from the restaurant domain, to provide us with underlying topics that the review sentences describe. To do this, we expand on our lexicons from Section 5.2, this time *automatically* aggregating lexicons for each of the five important restaurant attributes: *restaurant-type, cuisine, food, service,* and *staff* using Wikipedia[3] and DBpedia.[4] For example, to collect a lexicon of foods, we use domains ingredient and ingredientOf from the Food ontology in DBPedia,[5] querying the ontology using the SparQL interface (sample query shown in Figure 5.1). For cuisines, we sample from a list of cuisines on Wikipedia.[6]

We end up with 14 items for *restaurant-types* (e.g. "cafe"), 45 for *cuisines*

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

[3] https://www.wikipedia.org/

[4] http://wiki.dbpedia.org/

[5] Ontology: http://dbpedia.org/ontology/Food, Query Interface: https://dbpedia.org/sparql

[6] Cuisines: https://en.wikipedia.org/wiki/List_of_cuisines

(e.g. "Italian"), 4,913 for *foods and ingredients* (e.g. "sushi"), 12 for *staff* (e.g. "waiters"), and 2 for *service* (e.g. "customer service") [Oraby et al., 2017]. We also add lexicons for *ambience* (e.g. "decoration") and *price* (e.g. "cost") using vocabulary items from the E2E generation challenge [Novikova et al., 2017c]. Table 5.6 shows examples from our automatically-curated lexicons.

```
SELECT *
WHERE {
?fooda dbo:ingredient ?foodb .
?fooda rdfs:label ?namea .
FILTER (langMatches(lang(?namea), "en")) .

?foodb rdfs:label ?nameb .
FILTER (langMatches(lang(?nameb), "en")) .
}
LIMIT 10000 OFFSET 0
```

Figure 5.1: Sample DBPedia SparQL query to retrieve foods.

Restaurant	**Cuisine**	**Food**	**Staff**
restaurant	Italian	pizza	server
cafe	Mexican	chicken	waiter
cafeteria	French	jambalaya	barista
steakhouse	Asian	black beans	bartender
bistro	Japanese	fries	host
bar	Indian	sushi	busser

Table 5.6: Sample values for attributes in our restaurant lexicon.

It is critical to note here that the lexicons we curate are intended to cover a particular set of attributes and values, and that such a massive pool of reviews in fact contains *much more content* than we explicitly try to capture. Any given application will naturally define a set of specific attributes and values that it wants to communicate to the user (in our case, a dialog system aiming to describe restaurants, for example), and those would be the items it would look to annotate in the corpus.

5.3.2 Subsampling for Semantic Constraint

We note from our exploration of review sentences (as in Table 5.2), and adjective phrase descriptions (Table 5.4) that *food* items in particular elicit highly detailed and evaluative descriptions. We also note that the range of variation is *huge*, and in fact although our goal is to be able to train NNLG modules that can produce a diverse language, we need to subsample sentences from our reviews in order to enforce a kind of semantic constraint on the types of attributes that are described, and to attempt to find sentences that share a grounded semantic basis.

To enforce this kind of semantic constraints and "truth grounding" when selecting sentences, without severely limiting variability, we decide to focus on sentences that mention particular food values.[7] A pilot analysis of random reviews shows that some of the most commonly mentioned foods are *meats*: e.g. *"meat", "beef", "chicken", "crab",* and *"steak".* Beginning with the original set of over 4 million business reviews, we randomly sample a set of 500,000 sentences from restaurant reviews that mention of at least one of the meat items. Specifically, we end up with sentences spanning a range of 3k restaurants, 170k users, and 340k different reviews.

Table 5.7 shows examples of sentences that we identify as containing at least one *meat* value. The examples show how we also identify additional foods and other attribute types using our lexicons: for example, Row 2 shows 2 food values (toast and chicken) and one staff value (waiter), and Row 3 shows a restaurant value (chain restaurant). We also see the detailed descriptions we are interested in harvesting, such as *"tender and juicy"* and *"to die for"* in Row 4.

[7] We note that we also experimented with datasets without subsampling for semantic constraint, but noticed much more sparsity in the data, resulting in outputs that performed comparably more

1	We ordered the broasted <food>chicken</food>, <food>ribs</food> , and baked <food>potato</food> .
2	Our <staff>waiter</staff> recommended the french <food>toast</food> and the sage fried <food>chicken</food> benedict .
3	I had the rigatoni with <food>chicken</food> and <food>mushrooms</food> , and my husband had the <food>salmon</food> <food>marsala cream sauce</food> which was good at a <restaurant>chain restaurant</restaurant> .
4	The <food>chicken</food> pieces were tender and juicy , and the <food>rest noodles</food> sauce was to die for .
5	The <food>bbq skewers</food> were okay , but the <food>meat</food> was a little dry .

Table 5.7: Sample sentences containing food values grounded in *meat* descriptions.

5.3.3 An Algorithm for MR Creation

Given our set of 500k sentences containing at least one mention of a *meat* values (among others), we proceed to further characterize different semantic and stylistic properties for each sentence, in order to create a set of NL to MR pairs for training NNLGs.

Our NL to MR creation method is summarized in Algorithm 1. First, 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. Then, 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 5.2.

We identify all nouns and noun compounds, and search for them in the attribute lexicons, constructing *(attribute, value)* tuples if a noun is found in a lexicon, e.g. *(food, chicken-chimichanga)* in Figure 5.2. Next, for each *(attribute, value)* tuple, we extract all *amod*, *nsubj*, or *compound* relations between a noun value

poorly on semantic fidelity: thus we leave further exploration of this to future work.

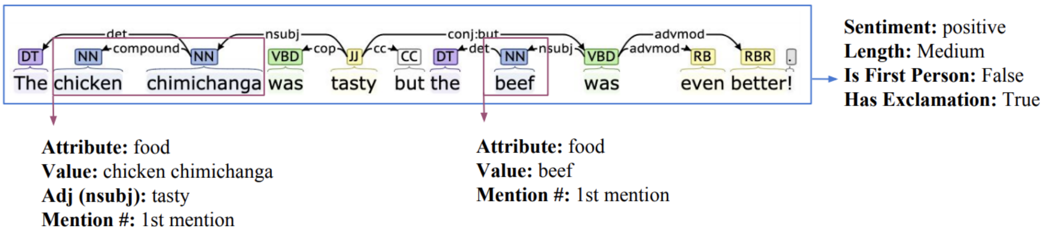


Figure 5.2: Extracting information from a review sentence parse to create an MR.

in the lexicons and an adjective using the dependency parse, resulting in *(attribute, value, adjective)* tuples. We add in *"mention order"* into the tuple, because some attributes are mentioned multiple times in the same reference.

We also collect sentence-level information. For *sentiment*, following the simplifying assumption in previous work on sentiment transfer [Shen et al., 2017], we tag each sentence with the sentiment inherited from the "star rating" of the original review it appears in. We bin the sentiment 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).[8]

We observe other stylistic points of interest in the data that we can easily capture at the sentence-level. To experiment with stylistic control of sentence length, we assign each sentence a *length* bin of *short* ($\leq$ 10 words), *medium* (10-20 words), and *long* ($\geq$ 20 words). Half of the sentences are in first person and around 10% contain an exclamation, both of which 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 other aspects of a hyperbolic style that we wish to capture.

[8] 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.

Algorithm 1: YelpNLG corpus creation.

```
Input: 300k review sentences;
Output: 300k MR-sentence pairs;
for sentence in sentences do
    parsed-sentence ← parse(sentence);
    values ← extractNN(parsed-sentence);
    dependencies ← extractDeps(parsed-sentence);
    SENT ← getReviewSentiment(sentence);
    LEN ← getLengthBin(sentence);
    FIRST-PERS ← isFirstPerson(sentence);
    EXCLAIM ← hasExclaimation(sentence);
    for val in values do
        ATTR = findValueInLexicons(val);
        if ATTR then
            ADJ ← getAdjectives(dependencies);
            MENTION ← getMention(sentence);
        MR[sentence] + = (ATTR, VAL, ADJ, MENTION);
    MR[sentence] + = (SENT, LEN, FIRST-PERS, EXCLAIM);
```

5.4 The YelpNLG Corpus

Thus, our method in Algorithm 1 gives us eight rich features that describe a combination of semantic and stylistic information about a given sentence: **attribute, value, adjective, mention order** on the attribute level, and **sentiment, length, first-person,** and **exclamation** at the sentence level. Using this information for each sentence, we construct an MR that consists of relational tuples for each attribute-value pair, as well as additional information for sentence-level features.

Using this method, we create the YelpNLG corpus: a set of 300k MR to NL realizations created using off-the-shelf tools and freely available data [Oraby et al., 2017, Oraby et al., 2019].[9] Table 5.8 shows sample sentences with the matching MRs that we create. We reiterate here that the attributes and values we use in our MRs only represent a subset of what could be selected to represent in the data: we focus specifically on attributes that can elicit interesting descriptions and evaluations.

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 5.8: Sample sentences and automatically generated MRs from YelpNLG. Note the style information that is marked up in these MRs compared to those in E2E or WebNLG.

In Row 1, we see the MR from the example in Figure 5.2, 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.

[9] The YelpNLG corpus is available at: https://nlds.soe.ucsc.edu/yelpnlg

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.[10] 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.

It is clear from our examples that our YelpNLG MRs not only include attribute-value pairs as in other datasets, but also use a relational tuple format to group together dependency relations for values, e.g. *(food, brioche-bun, yummy)*. We also include additional information describing stylistic features of the NL, characterizing sentiment, length, personal pronouns, and exclamations.

YelpNLG is also significantly larger and more stylistically diverse as compared to existing datasets. Table 5.9 compares YelpNLG to previous work in terms of data size, unique vocab and adjectives, entropy, and average reference length.[11] From the table, we can see that YelpNLG is over 5 times as large as the E2E dataset [Novikova et al., 2017c], with around 235k training instances, and has a vocabulary of 41k unique words, more than 15 times as large as the vocabulary in E2E, all without the need for any human crowdsourcing.

We also measure stylistic and structural variation in YelpNLG, in terms of simple contrast (markers such as "but" and "although"), and aggregation (e.g. "both" and "also") [Juraska and Walker, 2018], showing how our dataset is much

[10] 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 qualitatively later in this section.

[11] We described how we compute entropy in detail earlier, in our experiments in Section 4.4.1.

larger and more varied than previous work. We see that around 9% of YelpNLG references contain an explicit contrast marker, as compared to around 5% in E2E, and around 6% of them contain a aggregation marker, with less than 2% containing them in E2E. Only our PersonageNLG [Oraby et al., 2018b] corpus has a large percentage of aggregation, at 56%.

We note that the Laptop, E2E, and PersonageNLG 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.

	E2E	Laptop	PersonNLG	YelpNLG
Train Size	42k	8k	88k	**235k**
Train Vocab	2,786	1,744	224	**41,337**
Train # Unique Adjs	944	381	61	**13,097**
Train Entropy	11.59	11.57	9.34	**15.25**
Train RefLen	22.4	26.4	**28.33**	17.32
% Refs w/ Contrast	5.78%	3.61%	0%	**9.11%**
% Refs w/ Aggregation	1.64%	2.54%	**56.3%**	6.39%

Table 5.9: NLG corpus statistics from E2E [Novikova et al., 2017b], Laptop [Wen et al., 2016], PersonageNLG [Oraby et al., 2018b], and YelpNLG [Oraby et al., 2019].

Since our MRs are all retrofit, we examine the distribution of MR lengths that organically occur in YelpNLG. Figure 5.3 shows this distribution of MR length, in terms of the number of attribute-value tuples. We see that we naturally have a higher density of shorter MRs, with around 13k instances from the dataset contain around 2.5 attribute-value tuples, but that our MRs go up to 11 tuples in length. We can see some of these different MR sizes by referring back to Table 5.8, where Rows 1-2 each have 2 tuples, Row 3 has 3 tuples, and Row 4 is the longest shown, with 4 tuples in the MR.

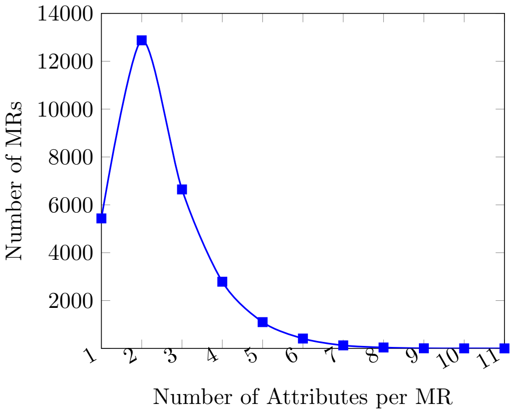


Figure 5.3: MR distribution in YelpNLG train.

We also conduct a small qualitative study to evaluate how well our YelpNLG MR to NL pairs are rated in terms of content preservation (how much of the MR content appears in the NL), fluency (how "natural sounding" the NL is), 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 6.4.3, but we first perform this experiment here to get a sense of our input data quality before using it to train a neural model.

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 an average 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 our NL, and that our NL are also fluent. This is an important check because correct grammar/spelling/punctuation is not a restriction in Yelp reviews, and we hope to capture sentences that make sense and flow fluidly. 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.

Although it is good to see that the YelpNLG corpus is stylistically diverse and rated reasonably well for content preservation, reference fluency, and correct sentiment, it is not without limitations. For one, our attribute labeling method is limited by our lexicons, which may be large for attributes like *food*, but certainly do not capture the full scope of attributes and values from within the reviews. Also, our MR creation relies on a successful parse to capture noun-phrases and associated adjectives, which is also error prone, potentially leading to missing and/or incorrectly tagged attributes, values, and adjectives. However, we believe that the size of the data given the trade-off between simplicity and speed of the curation method as opposed to costly, tedious crowdsourcing may makes it reasonable to accept some noise within the data as we explore whether we can use it to train NNLGs.

5.5 Summary

The problem of how to create corpora for language generation is long standing one in traditional NLG [Dale et al., 1998], but is significantly compounded in the case of neural NLG, where datasets must be *particularly large* in order to suffice for training neural models, and diverse enough to allow for varied language generation.

In order to acquire datasets large enough for NNLG, recent methods for collecting datasets have relied almost exclusively on crowdsourcing [Novikova et al., 2017b, Lebret et al., 2016, Gardent et al., 2017b, Wen et al., 2016, Wen et al., 2015], asking crowd workers to write a reference text given some input meaning representation. These efforts have frequently been in domains such as Wikipedia and restaurant description, where the communicative goal is mainly to inform the reader or listener of information pertaining to attributes like names and locations. For this reason, sentence constructions within these existing datasets are frequently dull and repetitive, as we showed in Table 5.1.

In this chapter, we described our novel approach to the corpus creation problem in NNLG, tackling both the problem of scalable corpora creation without crowdsourcing, and the problem of limited variation in the data. We use the observation that restaurant reviews are naturally highly descriptive, and that there is a massive amount of review data freely available online. Section 5.2 shows our pilot exploration of the descriptive language style used in restaurant reviews.

In Section 5.3, we presented our method to select suitable sentences from the restaurant review domain to use as references in our corpus, and show our algorithm to extract semantic and stylistic information from the references to automatically create meaning representations using off-the-shelf tools. Using our method, we create YelpNLG, a set of 300k MR-to-reference pairs in the restaurant review domain that can be used for NNLG. We show that YelpNLG is massively larger and more stylistically varied than any existing corpus for NNLG to date, as demonstrated in Table 5.9.

In the next chapter, we describe how we use the YelpNLG corpus to train neural models that, for the first time, have access to masses of varied semantic information and stylistic constructions, and explore how we can jointly control semantics and style in our NNLG outputs.

Chapter 6

Controlling Style in Neural NLG

6.1 Overview

Our ultimate goal in this thesis is to push the boundaries of controllable Neural Natural Language Generation (NNLG): to be able to train a neural generator to produce outputs that are as natural and human-like as possible. To this end, thus far in this thesis, we have shown how we produce simple style choices, such as aggregation operations and pragmatic marker choices, with a neural model trained using a synthetic corpus of restaurant descriptions, PersonageNLG, which we create based on the E2E NLG dataset in Chapter 4. In Chapter 5, we showed how we use off-the-shelf tools and resources to produce a corpus of user restaurant reviews, YelpNLG, without the need for any crowdsourcing or manual overhead. We also describe how YelpNLG is significantly larger, more varied, and more descriptive than any existing datasets for Natural Language Generation (NLG).

Given our experiments on introducing style supervision to NNLG systems, and armed with our YELPNLG corpus, we are now in the unique position of being able to explore whether we can generate neural model outputs from naturally occurring data, and whether we can jointly control the multiple interacting aspects of style that the data contains: namely, lexical choice, adjectival descriptions, length, sentiment, pronoun use, and exclamations.

In this chapter, we describe our experiments on the YELPNLG corpus, beginning with how we systematically create different versions of the corpus to test the effect of adding additional style information into our style encoding in Section 6.2. In Section 6.3.4, we present the three NNLG models we design for our experiments, building on our previous work on producing style in Chapter 4, each encoding increasing amounts of style information, and with novel changes to accommodate the enhanced relational structure of our Meaning Representation (MR)s in YELPNLG. Finally, in Section 6.4, we present a rigorous evaluation of our how each of our models performs on the tasks of semantic preservation and joint stylistic control of multiple interacting parameters.

6.2 Corpus Preparation

To recap the different types of style information in our YELPNLG MRs, we show an Natural Language (NL) review sentence and automatically-constructed MR from YELPNLG in Table 6.1. We also include the parse for the sentence in Figure 6.1, showing the basic dependencies which we use to construct the MR, as we described in Section 5.3.3.

We point attention to the different types of style information within the

MR: at the attribute-value level, relational tuples describe the relationships between attributes and values and their adjectival modifiers (`amod` relations from the dependency parse), as well as information on which reference of the attribute the tuple refers to (i.e. mention number). Additionally, at the sentence level, we include information on sentiment, length, whether or not the sentence is written in the first person, and whether or not the sentence is an exclamation.

Review Sentence NL: *With crispy skin and tender meat, the duck is perfect rolled up with some scallion and a little hoisin sauce.*

Automatically-Constructed MR:
(attr=food, val=tender-meat, adj=no-adj, mention=1), (attr=food, val=duck, adj=perfect, mention=1), (attr=food, val=scallion, adj=no-adj, mention=1), (attr=food, val=hoisin-sauce, adj=little, mention=1)
+[sentiment=positive, len=long, first-person=false, exclamation=false]

Table 6.1: A sample NL and MR pair from the YelpNLG corpus.

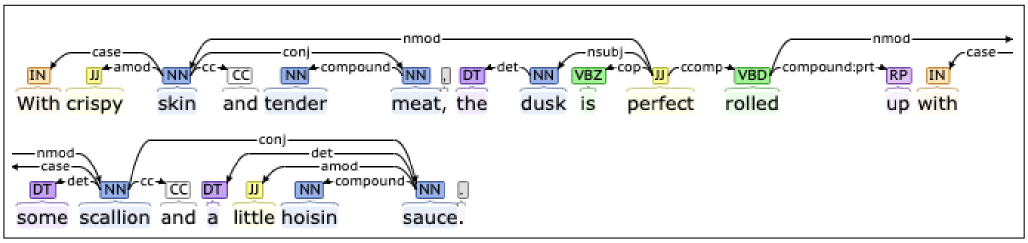

nmod
case
conj
amod
cc
compound
nsubj
det
cop
ccomp
compound:prt
nmod
case
IN JJ NN CC NN NN , DT NN VBZ JJ VBD RP IN
With crispy skin and tender meat, the dusk is perfect rolled up with
nmod
case
conj
det
amod
det
cc
compound
DT NN CC DT JJ NN NN .
some scallion and a little hoisin sauce.

Figure 6.1: Parse with basic dependencies for the sentence in Table 6.1.

Given the mix of semantic and style information encoded in our MRs, we are interested to experiment with what kind of output we can produce with neural models as we vary the amount of semantic and stylistic information they are provided. Thus, we decide to create *different versions* of our corpus with increasing amounts of style information: beginning with a version that contains only attributes and values, then one that contains the adjectival dependencies, followed by one that contains sentiment, and finally a version that contains *all* of the style information we have available in YelpNLG.

Thus, for each of the 300k MR-NL pairs in YelpNLG, we create 4 MR variations:

- Base: The MR contains only the basic content items, i.e. attributes like restaurant, food, cuisine, and their corresponding values from the lexicons.
- + Adj: The basic MRs, adding in any adjectives with a dependency relation to any of the basic MR content items.
- + Sent: Same as +Adj, but including a single attribute-value pair for sentiment.
- + Style: Same as +Sent, but adding in style information on mention order, what length it is, whether it is in the first person, and whether it contains an exclamation.

Table 6.2 shows another NL from YelpNLG, this time showing an example of each of the 4 MR variations we create. As we progress from Base, to +Adj, to +Sent, to +Style, we see how the MRs become richer, encoding more nuances of style information in the data.

6.3 Model Design

In this section, we recap our NNLG model design, based on the standard Sequence-to-Sequence (seq2seq) architecture we described in Chapter 3.[1] We add

[1] We note here that we also experimented with different architectures for NNLG, specifically the transformer model [Vaswani et al., 2017, Vaswani et al., 2018], but found that our outputs are comparable, thus we focus our discussion on the more heavily used seq2seq model in this thesis.

	I ordered the chicken with mushroom sauce, and the chicken was rubbery and hard to cut, and the gravy was also too salty.
BASE	(*attr*=food, *val*=chicken), (*attr*=food, *val*=mushroom_sauce), (*attr*=food, *val*=chicken), (*attr*=food, *val*=gravy)
+ADJ	(*attr*=food, *val*=chicken, *adj*=no-adj), (*attr*=food, *val*=mushroom_sauce, *adj*=no-adj), (*attr*=food, *val*=chicken, *adj*=rubbery), (*attr*=food, *val*=gravy, *adj*=salty)
+SENT	(*attr*=food, *val*=chicken, *adj*=no-adj), (*attr*=food, *val*=mushroom_sauce, *adj*=no-adj), (*attr*=food, *val*=chicken, *adj*=rubbery), (*attr*=food, *val*=gravy, *adj*=salty) +[*sentiment*=negative]
+STYLE	(*attr*=food, *val*=chicken, *adj*=no-adj, *mention*=1), (*attr*=food, *val*=mushroom_sauce, *adj*=no-adj, *mention*=1), (*attr*=food, *val*=chicken, *adj*=rubbery, *mention*=2), (*attr*=food, *val*=gravy, *adj*=salty, *mention*=1) +[*sentiment*=negative, *len*=long, *first_person*=true, *exclamation*=false]

Table 6.2: Sample of all 4 MR variations with increasing style information for a given NL.

details to describe how we encode the different levels of information in our richer YelpNLG MRs.

As we described in Section 3.3, in the standard Recurrent Neural Network (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 shown in Equation 6.1 [Klein et al., 2018].

$$p(w_{1:T}|x) = \prod_{1}^{T} p(w_t|w_{1:t-1}, x) \tag{6.1}$$

In the case of our YelpNLG MRs, the input $x_{1:S}$ is a sequence 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!"*

6.3.1 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 they assume. 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 Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997].

6.3.2 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 are sentence-level, specifically, 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 $x_{1:S}$ as shown in Equation 6.2 where f is the set of new feature constraints added to the model.

$$p(w_{1:T}|x) = \prod_{1}^{T} p(w_t|w_{1:t-1}, x, f) \tag{6.2}$$

We represent these additional features as a vector of additional supervision tokens or side constraints [Sennrich et al., 2016], similar to how we added in personality tokens in our work on PersonageNLG in Chapter 4 [Oraby et al., 2018b], and how other previous work has added domain encodings to the end of word embeddings in related tasks in machine translation [Kobus et al., 2017], or markers for contrast [Reed et al., 2018]. 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.

6.3.3 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. [Luong et al., 2015] using the *general* method of computing attention scores.

6.3.4 Model Configurations

To evaluate our models fairly, 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 precisely the same split.[2] As always, all models are trained using lower-cased and delexicalized reference texts.

We describe final model configurations for the most complex model, +STYLE in detail here, 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 with the lowest perplexity to evaluate on a test dataset. Beam search with three beams is used during inference. The values of MR attributes are represented using 300 dimensional embeddings. The target side word embeddings are initialized using pretrained Glove word vectors [Pennington et al., 2014] which get updated during training.

6.4 Evaluation

In this section, we provide a detailed set of quantitative and qualitative metrics designed to systematically evaluate how well each model adheres to the semantic and stylistic constraints we provide in its input MRs. We begin with a

[2] Since we randomly split the data, we compute the over- lap 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.

set of automatic evaluations targeted at semantics, followed by more novel evaluations targeting different aspects of style. We follow these with a qualitative human evaluation aimed at content preservation, fluency, and sentiment correctness.

Table 6.3 shows some 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), noting importantly here that the BASE, +ADJ, and +SENT models use the same MR minus the additional information. Row 1 shows an example of incorrect sentiment for BASE, and correct sentiment for the rest; +ADJ gets the adjectives right, +SENT is more descriptive, and +STYLE hits all targets.

In Row 2, we see another example of incorrect sentiment for BASE, and although +ADJ and +SENT both express the correct sentiment, in fact in interesting ways (*"sauce tasted like it was from a can"* and *"sauce was not even warm"*), the +STYLE model produces a more brief articulation in order to hit the medium length target, i.e. *"sauce had no taste"*.

Row 3 gives an example of extra length in +STYLE, *"the meat was so tender and juicy that it melted in your mouth"*. Row 4 shows another 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 move on here to our quantitative and qualitative analysis of the performance of each model, deferring more examples of the broad range of stylistic variation we can produce to later on in Section 6.6.

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, meat, chewy, mention=1), (food, sauce, no_adj, mention=1)** **+[sentiment=negative, len=medium, first_person=false, exclamation=false]**
	Base	The meat was tender and the sauce was delicious.
	+Adj	The meat was very chewy and the sauce tasted like it was from a can.
	+Sent	The meat was very chewy and the sauce was not even warm.
	+Style	The meat was chewy and the sauce had no taste.
3	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.
4	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 6.3: Sample test outputs for each model (only showing the +style MR).

6.4.1 Automatic Semantic Evaluation

Automatic Metrics. As in our evaluation of our models trained on PersonageNLG in Chapter 4, we begin with an automatic evaluation using standard metrics frequently used for machine translation, again using the scripts provided by the E2E Generation Challenge[3] 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 6.4 summarizes the results for BLEU, METEOR, CIDEr, and NIST, where higher numbers indicate better overlap (shown with the ↑). We again note that we include these metrics for completeness and for comparative purposes, as they are not well-suited to this task, since they are based on n-gram overlap which is not a constraint within our models.

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.

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

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 (of different types) [Reed et al., 2018]. We find counts of two types of common mistakes: deletes, where the model fails to realize a value from the input MR, and repeats, where the model repeats the same value more than once.[4]. Thus,

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

[4] We evaluate other semantic errors, i.e. substitutions and hallucinations, through our human evaluation of quality in Section 6.4.3, since they are difficult to automatically identify given the vast array of possible values in YelpNLG.

we compute SER 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 [Wen et al., 2015].

Table 6.4 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 6.4.3). 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 6.4 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.

6.4.2 Automatic Stylistic Evaluation

Achieving Other Style Goals. Here we compute stylistic metrics to compare the model outputs using various different tools, with results shown in Table 6.5. For **vocab**, we find the number of unique words in all outputs for each model. We compute average Flesch Reading Ease [Farr et al., 1951] for **readability**, which gives a score to each sentence, ranging from 0 (very complex) to 100 (very simple).[5] We find the average **output length** by counting the number of words, and average number of **adjectives per output** for each model. We compute Shannon text

[5] We use the python package, TextStat.

entropy as in Section 4.4.1. 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 ↑), except for readability, where a lower score means more complex sentences (indicated by ↓). Readability offers an interesting tradeoff between complexity and naturalness: while complex sentences (lower scores) are less "readable", they may be more structurally interesting.

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 5.9. For vocab, the simplest BASE model has the highest scores; the model with the largest amount of supervision, +STYLE, has the smallest vocab compared to the other models, since we provide the most constraints on word choice. BASE, the least constrained, has the most freedom in terms of word choice. BASE has the highest readability (i.e. the most structurally simple sentences, although the differences are small), which is interesting in conjunction with sentence length: as sentence length goes up (significantly) in the +STYLE model, readability goes down slightly (i.e. sentences are more structurally complex). +STYLE also has the highest average adjectives and entropy. 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), 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. We explore this variation in construction later in Section 6.5.

			Base	+Adj	+Sent	+Style
1	Vocab	↑	**8,627**	8,283	8,303	7,878
2	Readability	↓	71.11	70.62	70.08	**70.01**
3	Len	↑	11.27	11.45	11.30	**13.91**
4	Adj/Op.	↑	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 6.5: Automatic stylistic evaluation metrics (higher is better). Paired t-test BASE vs. +STYLE all $p < 0.05$.

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 NNLG 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 for LEN=LONG is 22.74, very closely matching the 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).

6.4.3 Human Quality Evaluation

We evaluate output quality using human annotators on Mechanical Turk, similarly to how we evaluated the quality of our corpus itself in Section 5.4. We randomly sample 200 MRs from the test set, along with the corresponding outputs for each of the 4 models. We ask 5 annotators to rate each output on a 1-5 Likert scale for **content**, **fluency**, and **sentiment** (1 for very negative sentiment, 5 for very positive[6]). Figure 6.2 shows the HIT interface we use.

The fluency measure aims to detect both grammatical errors and problems

[6] As in Sec 5.4, we scale the sentiment scores into 3 bins to match our Yelp review sentiment.

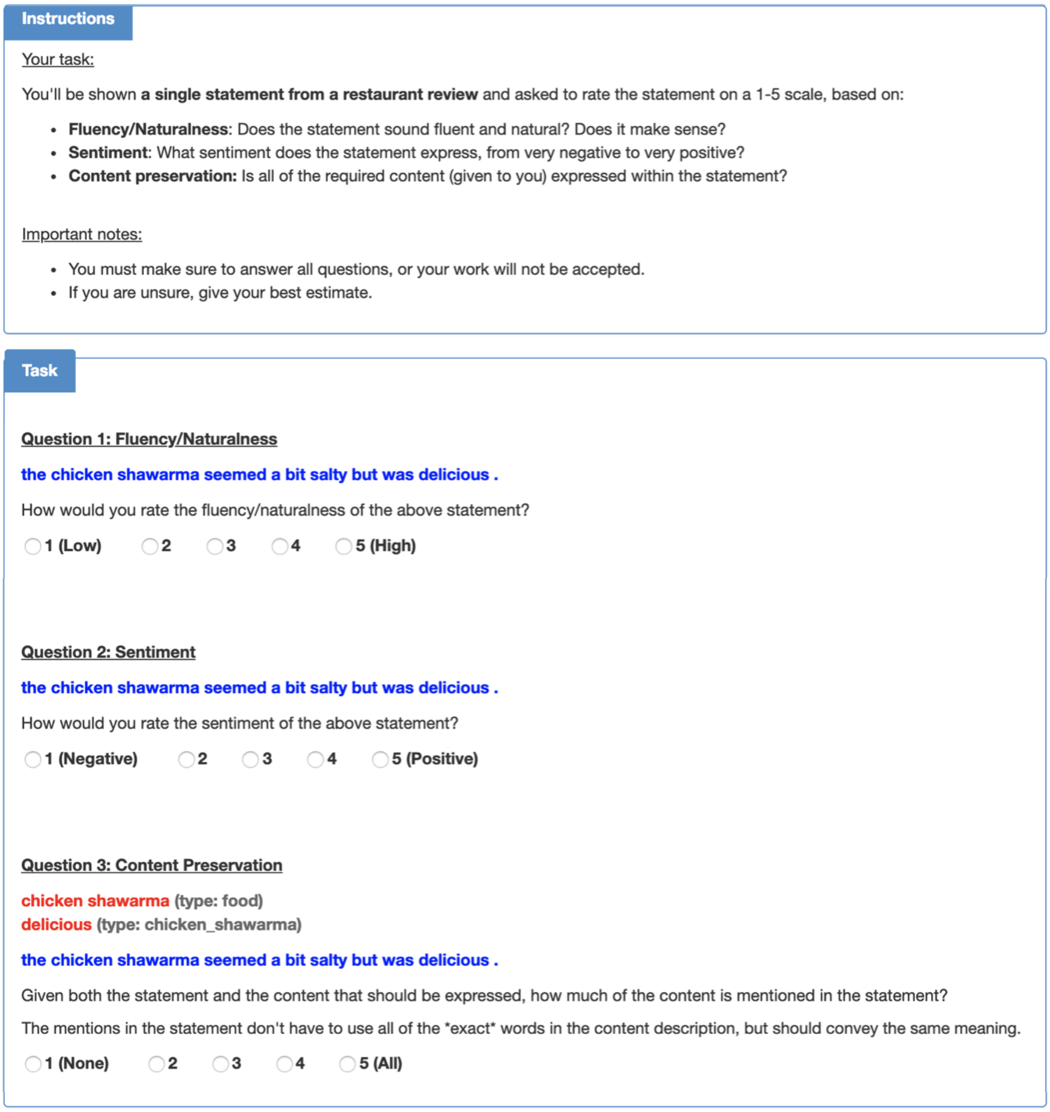
with general fluency. For content and fluency, we compute the average score across all 5 raters for each item, and average those scores to get an average rating for each model, such that higher content and fluency scores are better. For sentiment, we again compute the average rating for all 5 annotators per item, but instead we compute the *difference* between their average score and the true sentiment rating in the reference text (from the original review), so that lower sentiment error is better. Table 6.6 shows the average score out of 5 for each criteria and model.[7]

For content and fluency, all average ratings are very high, above 4.3. The differences between models are small, but it is interesting 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 Error	↓	0.75*	0.71*	0.67*	0.56

Table 6.6: 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 $p < 0.05$, where +STYLE is significantly better.

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



Your task:

You'll be shown **a single statement from a restaurant review** and asked to rate the statement on a 1-5 scale, based on:

- **Fluency/Naturalness**: Does the statement sound fluent and natural? Does it make sense?
- **Sentiment**: What sentiment does the statement express, from very negative to very positive?
- **Content preservation:** Is all of the required content (given to you) expressed within the statement?

Important notes:

- You must make sure to answer all questions, or your work will not be accepted.
- If you are unsure, give your best estimate.

Question 1: Fluency/Naturalness

the chicken shawarma seemed a bit salty but was delicious .

How would you rate the fluency/naturalness of the above statement?

○ **1 (Low)** ○ **2** ○ **3** ○ **4** ○ **5 (High)**

Question 2: Sentiment

the chicken shawarma seemed a bit salty but was delicious .

How would you rate the sentiment of the above statement?

○ **1 (Negative)** ○ **2** ○ **3** ○ **4** ○ **5 (Positive)**

Question 3: Content Preservation

chicken shawarma (type: food)
delicious (type: chicken_shawarma)

the chicken shawarma seemed a bit salty but was delicious .

Given both the statement and the content that should be expressed, how much of the content is mentioned in the statement?

The mentions in the statement don't have to use all of the *exact* words in the content description, but should convey the same meaning.

○ **1 (None)** ○ **2** ○ **3** ○ **4** ○ **5 (All)**

Figure 6.2: MTurk HIT for YelpNLG human evaluation.

6.5 Language Template Variation Analysis

Since our testset 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 templates, we delexicalize each instance 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 6.3 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 30k). 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.

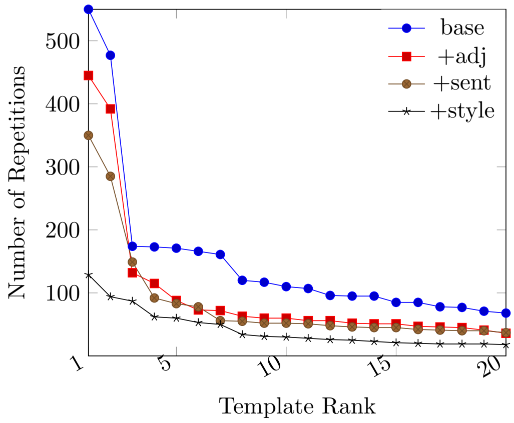


Figure 6.3: Number of output template repetitions for the 20 most frequent templates (+STYLE has the fewest repetitions, i.e. is most varied).

Table 6.7 shows examples of the templates generated 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; for each model, we show the top 8 most repeated templates, followed by a sample of 8 "rare" templates. Note for example that the largest number of reps is only 550 even for the BASE model, which in general contains more repetition, meaning that the models mostly generate novel outputs for each test instance.

As we look at the rare templates for each model, it is interesting to see the types of unique constructions and phrases that are produced. For example, for BASE, we see comparisons like *"[FOOD] was good, but not as good as [FOOD]"*, and even an example of a elaboration to describe a food in more detail with a restricted relative clause, *"i had the [FOOD]... which is a [FOOD] with..."*. For the +STYLE model, we see many examples of aggregation and interesting descriptions *"the [FOOD] was flavorless, the soy [FOOD] was watery and the [FOOD] tasted like it came from a can"*, as well as contrast, *"[FOOD] was nice and crispy, but i could have had a little more"*.

6.6 More Examples: A Vast Array of Stylistic Variation

In terms of quantitative evaluation, we have shown that our +STYLE model has produces the most variable output while adhering closely to semantic requirements. We have also shown that it has performed well on human quality evaluations, and produces more novel template than the less-informed models. In this section, we focus on the +STYLE model, showing different examples of rich stylistic variation that it is able to produce.

# 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.*
8	*i had the [FOOD] fried [FOOD] and eggs.*
7	*i had the [FOOD] and [FOOD] platter.*
6	*the [FOOD] [RESTAURANT] was good, but the [FOOD] was a little dry.*
5	*i asked the [STAFF] if the [FOOD] was cooked to order.*
5	*the [FOOD] was good quality [FOOD], but not great.*
1	*the [FOOD] wings were good, but not as good as the [FOOD] coating [FOOD] [FOOD].*
1	*i ordered the [FOOD] [FOOD] and the [FOOD] was dry and the buns were soggy.*
1	*i had the [FOOD] [FOOD] inside [FOOD] , which is a [FOOD] with pineapples and [FOOD].*

	+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.*
8	*the [FOOD] was tender and the [FOOD] was melted in your mouth.*
4	*the [FOOD] in my [FOOD] tasted like it came out of a can.*
4	*the [FOOD] in my [FOOD] tasted like it had been sitting out for hours and it was so dry.*
4	*i ordered a [FOOD] with [FOOD], and it was just ok, nothing to write home about.*
1	*the [FOOD] was flavorless, the soy [FOOD] was watery, and the [FOOD] tasted like it came from a can.*
1	*i ordered a [FOOD] dish and the [FOOD] was so dry it was hard to eat.*
1	*the grilled [FOOD] boneless [FOOD] breast was delicious with the [CUISINE] spices.*
1	*the [FOOD] of the [FOOD] was nice and crispy, but i could have had a little more.*

Table 6.7: Top 8 “most repeated” and sample 8 “rare” templates from BASE and +STYLE.

6.6.1 Personal Pronouns

We begin our discussion of the different examples of stylistic variation by looking at examples of personal pronoun use in our outputs, which may offer a more personal and natural way to convey the intended content [Boyce and Gorin, 1996]. In this case, we provide Model +STYLE with a boolean value in the MR indicating whether or not it should use "first-person pronouns" in the realization. This means that if *first-person=true*, the model is expected to use a first person pronoun, but in the case of *first-person=false*, there is no explicit direction as to what pronouns to use.

As a result, we observe that Model +STYLE makes various different pronoun choices. Specifically, within our Model +STYLE test outputs, we find around 17k occurrences of first-person pronouns used within the 30k test set (e.g. *i, me, we, us*), as compared to 151k occurrences in training (within the 235k train set).

We exemplify some of the pronoun choices the model makes in Table 6.8. Rows 1-3 of the table show examples of *first-person=true*: Row 1 mentions the speaker themselves, and their husband, *"I had the fish meat and it was very good, and my husband had the chicken ... which he loved"*; Rows 2-3 satisfies the first person constraint by referencing the speaker within a group, *"the waiter was friendly and helpful to us"*, and *"we had the ..."*. Rows 4-5 are set to *first-person=false*, and we see pronouns "she" and "he" used.

6.6.2 Contrast

The rest of the examples of stylistic variation we describe are ones that the model is *not explicitly instructed to perform* in the MR. We see many examples of

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 6.8: Examples of different pronouns from Model +STYLE.

contrast in the Model +STYLE outputs, which we quantify using a basic heuristic by counting occurrences of contrast markers, (e.g. *"but"*, *"however"*, *"despite"*, *"although"*) in training and test, and find that around 2.7k occurrences within the 30k test set, compared to 21k occurrences in training (within the 235k train set).

Table 6.9 shows examples of contrast in the outputs of Model +STYLE. From the table, we see contrasts with operator "but", serving different purposes in the realizations. In Row 1, the realization begins by describing the food value *chicken* (correctly realizing it 3 times), followed up with an evaluative expression, *"but the chicken is a little on the dry side"*. Row 2 instead begins with an evaluation, *"they were pretty good"*, then contrasts this with an interesting idiom, *"but nothing to write home about"*. Row 4 shows an example of an evaluation of one food item, *"rice was good"*, followed up with a contrasting phrase describing a speaker desire, *"but I wish there was more meat in the dish"*.

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*=positive, *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 6.9: Examples of contrast from Model +STYLE.

6.6.3 Aggregation

In terms of aggregation, we follow the same method we used to quantify contrast, counting occurrences of simple aggregation markers, (e.g. *"both"* and *"also"*) in training and test. In terms of these very basic (and lexically defined) keywords, we find around 1.1k occurrences within the 30k test set, compared to 15k occurrences in training (within the 235k train set).

Table 6.10 shows different types of aggregation that the model is able to produce, many of which are more complex than we can capture by simply looking for keywords. Recall that the model *is not explicitly instructed aggregate*, but organically produces constructions with aggregation while hitting other style goals.

In Row 1, the model aggregates all of the food items into a single list, then uses "and" operator to add in the *price* description. Rows 2-3 aggregate the evaluation *"very good"* and *"delicious"* for the two food items. In Row 4, we see aggregation with ellipsis: all food values are first aggregated in a list (appending *"etc."*, then the evaluation *"all of it was delicious!"* is applied to all items in the list.

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 6.10: Examples of aggregation from Model +STYLE.

6.6.4 Hyperbole

We also observe exciting examples of hyperbolic language in the model outputs in Model +STYLE, which again are not explicitly elicited by our MRs (only implicitly through adjective tagging). We construct a set of hyperbole markers by manually filtering a list of commonly occurring adjectival and adverbial phrases from training (both unigrams and bigrams), and count the number of instances that contain these markers in both our YelpNLG training data and +style model outputs.

Table 6.11 shows these counts for each marker, as a way to quantify the types of hyperbolic variation our model is able to learn organically from the data. From the table, we see that many of the markers are used very frequently by Model +style: for example, the most frequent unigram, *delicious*, is used around 3k times in the 30k outputs (as compared to around 8k times in the 235k training set), and the most common bigram, *very good*, is used 1.3k times in the outputs (2.8k times in training). We also note that some of the less common bigrams from training are never reproduced by the +style model, including *so delicious*, *so tasty*, and *really great*, although the less common unigrams are produced, e.g. *spectacular* and *terrific*. This implies that better encoding of more complex adjectival phrases in our MRs may be an interesting addition for future work.

Examples of other interesting hyperbolic output from Model +style are shown in Table 6.12. In Row 1, value *meat* is assigned adjective *spectacular*, then the value sauces is described with a wonderful use of hyperbole, *"to die for"*. We see other examples in the table, some required in the MR, and some generated organically by the model: *heavenly* and an interesting contrast *"a nice touch but not overpowering"* in Row 2, *"the best i have ever had"* in Row 3, and *"phenomenal"* in Row 4, all using valenced language that is novel to outputs from state-of-the-art NNLG systems, and is learned from rich the training data as the model tries to satisfy style goals such as sentiment and length.

Hyperbolic N-gram	Train Count (out of 235k)	Output Count (out of 30k)
UNIGRAMS		
delicious	8892	3223
perfect	6177	2152
amazing	4642	392
excellent	2864	264
awesome	2099	148
huge	2016	415
super	1986	80
fantastic	1335	110
wonderful	975	54
outstanding	547	35
incredible	515	39
succulent	305	26
fabulous	298	18
phenomenal	295	25
superb	232	21
delightful	215	18
exceptional	210	13
massive	187	17
spectacular	171	7
terrific	118	2
BIGRAMS		
very good	2852	1339
really good	1880	46
so good	1088	37
very tasty	942	138
so tender	437	113
do delicious	416	1
very flavorful	370	37
very nice	282	8
very fresh	264	8
absolutely delicious	227	0
so tasty	208	1
really tasty	200	0
very delicious	195	0
perfectly seasoned	184	16
so many	171	0
so fresh	140	3
so flavorful	123	5
so juicy	110	7
very moist	106	3
really great	98	0

Table 6.11: Counts of frequently occurring hyperbolic n-grams in YelpNLG training data and +STYLE model outputs (sorted by training counts).

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 6.12: Examples of hyperbole from Model +STYLE.

6.7 Summary

In this chapter, we presented our experiments on jointly controlling the multiple aspects of style marked up in our YelpNLG corpus. In Section 6.2, we describes the 4 different sets of MRs we create from YelpNLG, with increasing levels of style markup. Our BASE MRs only encode attribute and value information as tuples, +ADJ MRs add in adjectives to the attribute-value tuples, +SENT MRs include sentiment, and finally our richest MRs, +STYLE, add in information on the order attributes are mentioned, as well as sentence-level information, namely sentence length, use of personal pronouns, and whether or not the sentence contains an exclamation.

We described our experiments in training 4 different models using our set of MR variations in Section 6.3.4, presenting how we encode the different types of features for each model. We present our evaluation of the models' performance with a detailed set of semantic and stylistic evaluations in Section 6.4, including various automatic metrics such as readability and entropy. In general, we find that with our most detailed style markup, our +STYLE model is able to produce the most varied outputs in terms of our stylistic measures, and in fact does the best job in terms of lexical measures such as BLEU and METEOR. In terms of semantics, all of our models perform well, with under 10% semantic error rate across the board; and even though the simplest BASE model is naturally the best at semantic fidelity (with a 5% error rate), the +STYLE model only has a slightly higher error rate of 9%, a small sacrifice given the amount of variation in the outputs it produces, consistently hitting required style targets.

In Section 6.5, we showed an analysis of model outputs when templatized, as a way to assess the variety the models are able to produce. We find that Model +STYLE produces the most distinct templates, but that all models in general do not frequently reuse templates (i.e. most constructions produced are novel). In Section 6.6, we showed various different examples of stylistic variation that Model +STYLE produces: specifically in terms of interesting operations such as personal pronoun use, contrast, aggregation, and hyperbole. We find that even without being explicitly instructed in the MR, Model +STYLE organically produces a vast array of interesting style choices, as guided by the variety in the data.

Chapter 7

Conclusions

7.1 Overview

State-of-the-art Neural Natural Language Generation (NNLG) models introduce a powerful new paradigm shift in Natural Language Generation (NLG), learning from data in an End-to-End (E2E) way, allowing us to generate text from input content without needing to hand-craft rules to guide the generator in expressing the required communicative goal. However, despite the prevalence of such claims of low-effort, fully data-driven generation from NNLG systems, it is unclear from recent work that the E2E architecture employed by these systems is in fact capable of producing *controllable* semantics and style as was possible with Statistical Natural Language Generation (SNLG) systems.

In this thesis, we have described our own methods for filling the style gap in NNLG: creating datasets and models for controllable language generation. In this final chapter, we summarize our contributions, describe applications and limitations of this work, and provide directions for future work.

7.2 Contributions

In our introduction in Chapter 1, we framed the limitations of NNLG systems as research questions around three critical bottlenecks: style, data, and control, supporting our arguments through a review of current work in Chapter 2, and a deep-dive into the state-of-the-art in NNLG in Chapter 3. The objective of this thesis is to address these bottlenecks through controllable NNLG. We seek to develop NNLG systems that can produce output that both satisfies the required semantics as defined by an input Meaning Representation (MR), *and* simultaneously includes interesting and diverse structural and stylistic constructions. In this section, we describe our contributions in this thesis based on our original research questions targeting these NNLG bottlenecks.

7.2.1 The Style Bottleneck

Can we develop a supervision mechanism to produce style in NNLG models?

In Chapter 4, we presented our first set of experiments for inducing structural and stylistic variation into a state-of-the-art NNLG pipeline.

To create this experimental setting for controllable NNLG, we begin with data from the E2E generation challenge dataset [Novikova et al., 2017b], and use a statistical generator, PERSONAGE [Mairesse and Walker, 2010], to generate controlled variations of the E2E data based on the Big-Five personalities, with predefined style choices governing aggregation operations and pragmatic marker usage. In this way, we synthetically design the PERSONAGENLG corpus: a set of 88k MR to Natural Language (NL) utterances in five different personality styles.

The PersonageNLG corpus provided us with a controlled environment for testing whether an NNLG model can learn to produce both the required content *and* style for a given instance, where personality is a proxy for a multitude of different style choices, designed around aggregations operations and pragmatic marker usage. Given the corpus, we presented a series of experiments exploring how different amounts of supervision and input representations affect model outputs.

We tried two different methods for supervision. First, we experimented with using a single token to identify which personality style to produce given an input MR, with Model_Token. Secondly, we experimented with a more detailed form of supervision with Model_Context, where we provided more detailed style parameters from Personage to the model with a change to the architecture, dictating more explicitly what choices the model should make, such as whether to include a particular hedge or pragmatic marker.

We compared each supervision method to a vanilla model that does not use any style encoding, and evaluated our models' performance through a series of quantitative and qualitative evaluations, including automatic metrics popular for evaluating NNLG systems, and our own set of error metrics designed to better describe the types of errors NNLG models are notorious for making: deletions, repetitions, insertions, and substitutions.

In our evaluations, we showed that while the vanilla model made the fewest semantic errors, the outputs loses any distinctive stylistic variation. With Model_Context, however, we were able to achieve our goal: we could *both* produce stylistically varied outputs that correlated with the required personalities, *and* preserve semantic fidelity with notably few errors [Oraby et al., 2018b], with a maximum of 8% semantic errors per test instance. We also showed that our model outputs were rated highly on naturalness through a human evaluation (all above 4.3/7). We also presented an experiment on generating outputs that combined features of multiple personalities with a novel model, MODEL_MULTIVOICE, a setting never seen in training, showing the model was able to interpolate between personality parameters, and generate novel outputs completely unlike what was seen in training [Oraby et al., 2018a].

7.2.2 The Data Bottleneck

Can we create sufficiently large and varied datasets to train NNLGs?

We tackled the data bottleneck in NNLG in Chapter 5 by developing a novel, scalable method for creating massive data-to-text corpora for training neural generators, using exclusively off-the-shelf tools and freely available data, and without the need for any crowdsourcing. Our goal here was not only to avoid having to do any crowdsourcing to collect sufficiently large datasets for NNLG, but also to be able to explore the masses of freely available review data online, full of rich, descriptive language grounded in peoples' experiences [Oraby et al., 2017, Oraby et al., 2019].

With these goals in mind, we presented our method to systematically "retrofit" an MR from an NL using entirely off-the-shelf tools, including part-of-speech and dependency information from the sentence parse, and entity-type information from open-source ontologies such as DBPedia. We curated lexicons intended to cover a particular set of attributes and values, but noted that the user reviews we used contain *much more content* than we can explicitly try to capture, depending on our intended use-case: any given application will naturally define a set of specific attributes and values that it wants to communicate to the user, and those would be the items it would look to annotate in the corpus.

Our method for dataset collection is robust to changes in application requirements, and the richness of the MRs generated is flexible. Additionally, datasets collected in this way are not bound by size: thus, with access to potentially massive amounts of data, a model should be able to learn to smooth out content that is extraneous to the goals of its application.

Ultimately in this thesis, to tackle the data bottleneck, we presented two of the largest and most stylistically varied corpora in NNLG, each complete with style markup useful for describing the references in a way that can be used to guide style generation in E2E frameworks:

1. PersonageNLG: A corpus of 88,000 MRs to reference texts based on the E2E Generation Challenge dataset and corresponding to stylistic properties of Big Five Personalities, including a variety of marked aggregation and pragmatic marker operations. This corpus provides a testbed for controllable NNLG, and has lead to new work on sentence planning and architectural changes for better model control [Reed et al., 2018, Harrison et al., 2019, Reed et al., 2019].
2. YelpNLG: A corpus of 300,000 MRs to reference texts created using freely available restaurant reviews from Yelp. MRs include semantic information as well as a novel characterization of descriptive lexical choice, sentiment, length, personal pronouns, and exclamations. YelpNLG is also significantly larger and more stylistically diverse as compared to existing datasets: it is over 5 times as large as the E2E dataset [Novikova et al., 2017c], with around 235k

training instances, and has a vocabulary of 41k unique words, more than 15 times as large as the vocabulary in E2E, all without the need for any human crowdsourcing.

7.2.3 The Control Bottleneck

Can we jointly control multiple interacting aspects of style with NNLGs?

Given our experiments on style control for NNLG based on our PersonageNLG, and armed with our massive and stylistically varied YelpNLG corpus, we presented the first experiments in NNLG style generation that show how to jointly control multiple interacting aspects of style in language: personality-based style in the restaurant description domain in Chapter 4, and multiple stylistic features including descriptive lexical choice, sentiment, and length in the restaurant review domain in Chapter 6.

Specifically for the YelpNLG corpus, we presented experiments on how to encode these style features in a state-of-the-art neural generation framework, and present a set of 3 different models with varying levels of semantic and stylistic encoding, based on the style markup in the YelpNLG corpus.

To evaluate joint preservation of semantics and style in our outputs, we presented rigorous evaluations for each of our models, including new metrics comparing model outputs based on vocabulary size, readability, sentence length, adjectives, entropy, contrast, and aggregation. In general, we found that with our most detailed style markup, our +style model was able to produce the most varied outputs in terms of our stylistic measures, and in fact did the best job in terms of lexical measures such as BLEU and METEOR.

In terms of semantics, all of our models performed well, with under 10% semantic error rate across the board; and even though the simplest BASE model was naturally the best at semantic fidelity (with a 5% error rate per MR, on average), the +STYLE model only had a slightly higher error rate of 9%, a small sacrifice given the amount of variation in the outputs it produces, consistently hitting required style targets. Model +STYLE also consistently outperformed the others in terms of the stylistic measures, consistently hitting style goals such as pronoun use (99% of the time) and exclamation (100% of the time). Similarly, we showed that all of our model outputs were rated competitively by human judges for content and fluency, with all ratings above 4.3/5.

We also presented a detailed analysis of the diverse stylistic variation in our model outputs. We find that Model +STYLE produces the most distinct templates, but that all models in general do not frequently reuse templates (i.e. most constructions produced are novel). In Section 6.6, we showed various different examples of stylistic variation that Model +STYLE produces: specifically in terms of interesting operations such as personal pronoun use, contrast, aggregation, and hyperbole. We find that even without being explicitly instructed in the MR, Model +STYLE organically produces a vast array of interesting style choices, as guided by the variety in the data. Our experiments showed that with a high descriptive corpus and some model supervision, we can jointly control stylistic features and produce a large array of new stylistic constructions, all without sacrificing semantic fidelity.

7.3 Applications

The use of personal assistant dialog systems such as Amazon Alexa or Google Assistant is on the rise. While these systems are currently able to help people do simple tasks such as setting alarms, saving reminders/shopping lists, and looking up basic information, there is rapidly increasing interest in enabling them to hold intelligent conversations as aids to children or the elderly, or guide users in complex decision making processes, such as finding a restaurant or choosing and booking a vacation. In order to do this, personal assistants need to be able to use natural language similar to humans, adapting their style based on the conversational context, the domain, and the needs of the user.

Thus, natural language generators for task-oriented dialog systems should be able to vary the style of the output utterance while still effectively realizing the system dialog actions and their associated semantics [Oraby et al., 2018a]. For example, if a system is able to control sentence planning operations such as sentence scoping, this affects the complexity of the sentences that compose an output, allowing the generator to produce simpler sentences when desired that might be easier for particular users to understand. Discourse structuring is often critical in persuasive settings such as recommending restaurants, hotels or travel options [Scott and de Souza, 1990, Moore and Paris, 1993], in order to express discourse relations that hold between content items [Stent et al., 2002]. Previous work has explored the effect of first person sentences on user perceptions of dialog systems [Boyce and Gorin, 1996]. Pragmatic variation, such as the use of different pragmatic markers or discourse cues, and lexical choice, or which words to use to express concepts, also have a clear effect on the perceived style of the output.

As we have described, stylistic control of many of these operations is the ultimate goal of this thesis, and through our work we have presented methods to develop end-to-end models that can learn in a data-driven way from real user data, paving the way for more work on making NNLG modules in dialog systems more natural and engaging.

7.4 Limitations

Here, we summarize some of the important limitations of our work:

- In the context of our experiments creating and using the PersonageNLG corpus for our first experiments on style control, we used a synthetically designed corpus, with predefined style operations. While using synthetic data for different language tasks is not uncommon [Weston et al., 2015, Dodge et al., 2015], it means that our outputs are inherently less natural than human-written ones, and that our models are limited in terms of the types of style choices they are exposed to and expected to produce.
- Our corpus creation method for YelpNLG, although completely avoiding any crowdsourcing and using exclusively off-the-shelf resources, is also inherently noisy. For one, our attribute labeling method is limited by our lexicons, which may be large for attributes like *food*, but certainly do not capture the full scope of attributes and values from within the reviews. Also, our MR creation relies on a successful parse to capture noun-phrases and associated adjectives, which is also error prone, potentially leading to missing and/or incorrectly tagged attributes, values, and adjectives. Although we hope that the learned language model will manage to smooth out noise in the data, this is still an

important limitation.

- In our exploration of joint control of semantics and style in YelpNLG, we use a simple method of including the style features as semantic constraints in our model [Sennrich et al., 2016]. While this method proved successful in allowing our models to learn the required style parameters, if we wish to vastly expand the number of parameters available to the model, we may need to encode them more concisely, or make architectural changes to accommodate more contextual information. Also, given that the strength of NNLG models is in their ability to learn in an E2E fashion directly from data, there is a trade-off to consider between control through increased supervision, and completely data-driven learning.

- We note that although we are able to generate outputs that are significantly more diverse than those of previous work, we still get a significant drop in variation as compared to training: for example, we have an average vocabulary of around 8k for our YelpNLG models, down from around 13k in the test references. This is explained by the "frequentist" approach that NNLG models generally employ, learning only the simplest and most prevalent way to realize the required content from training, and warrants more work on how to learn to produce rarer vocab items in the training data.

- Finally, although we explore the use of many new and interesting metrics for evaluating semantic and stylistic quality in our output, including semantic error rates, vocabulary, sentence complexity, and entropy, there is still a need for a suite of standard metrics in NLG [Novikova et al., 2017a].

7.5 Future Work

For future work, we are interested in improving our methods in terms of more large-scale, less noisy data curation methods, and improved style control. In terms of data curation, we are interested in reproducing our corpus generation method on various other highly-descriptive user review domains to allow for the creation of numerous useful datasets for the NNLG community. Similarly, we believe an important next step is to work on better quality control for datasets curated automatically using our method.

Secondly, in terms of methods for style control, we are interested in developing new 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. This may be particularly useful in the context of an application such as a dialog system, where more control over specific aspects of style or semantics may be necessary. In this kind of a setting, it may be useful to have a heavily detailed MR at generation time to produce a very specific and controlled realization. To this end, we are interested in including richer, more semantically grounded information in our MRs, for example using a form of Abstract Meaning Representations (AMR) [Dorr et al., 1998, Banarescu et al., 2013, Flanigan et al., 2014], and on including more explicit syntactic structure information from the sentence parses, as in Iyyer et al.'s work on syntactically controlled paraphrases [Iyyer et al., 2018].

Similarly, we are interested in trying other models currently being used on different sequential problems, such as the Transformer model [Vaswani et al., 2017, Vaswani et al., 2018]. We would also like to explore novel architectural changes to accommodate increased style encoding in our models, such as adding more supervision specifically to the decoder [Harrison and Walker, 2018], and further explore how to standardize metrics for evaluating NNLG models at scale.

Finally, in the context of practical applications, we would like to explore the use of NNLG systems like our own as functional NLG modules in state-of-the-art dialog systems. This will introduce interesting new engineering challenges, as well as the need to develop new evaluation metrics around quality control for interaction with real users. We believe that through more exploration of how users engage with dialog systems aimed at providing interesting and engaging experiences, we will be able to push the boundaries of NLG: towards systems that can begin to mimic the rich complexity of our own human expression.

Bibliography

[Abadi et al., 2016] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, pages 265–283, Berkeley, CA, USA. USENIX Association.

[Agarwal et al., 2018] Agarwal, S., Dymetman, M., and Gaussier, E. (2018). Char2char generation with reranking for the E2E NLG challenge. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 451–456, Tilburg University, The Netherlands. Association for Computational Linguistics.

[Angeli et al., 2010] Angeli, G., Liang, P., and Klein, D. (2010). A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512. Association for Computational Linguistics.

[Aubakirova and Bansal, 2016] Aubakirova, M. and Bansal, M. (2016). Interpreting neural networks to improve politeness comprehension. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2035–2041, Austin, Texas. Association for Computational Linguistics.

[Bahdanau et al., 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

[Banarescu et al., 2013] Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. (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.

[Bangalore and Rambow, 2000] Bangalore, S. and Rambow, O. (2000). Exploiting a probabilistic hierarchical model for generation. In *Proc. of the 18th Conference on Computational Linguistics*, pages 42–48.

[Bangalore et al., 2000] Bangalore, S., Rambow, O., and Whittaker, S. (2000). Evaluation metrics for generation. In *Proceedings of the First International Conference on Natural Language Generation - Volume 14*, INLG '00, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Barzilay, 2003] Barzilay, R. (2003). Information fusion for multidocument summarization: Paraphrasing and generation. Technical report, New York, United States of America.

[Barzilay and Lapata, 2005] Barzilay, R. and Lapata, M. (2005). Collective content selection for concept-to-text generation. In *Proceedings of the conference on*

Human Language Technology and Empirical Methods in Natural Language Processing, pages 331–338. Association for Computational Linguistics.

[Barzilay and Lapata, 2006] Barzilay, R. and Lapata, M. (2006). Aggregation via set partitioning for natural language generation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 359–366. Association for Computational Linguistics.

[Belz and Reiter, 2006] Belz, A. and Reiter, E. (2006). Comparing automatic and human evaluation of nlg systems. In *EACL*.

[Bengio et al., 2003] Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

[Bengio et al., 1994] Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

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

[Brown and Levinson, 1987] Brown, P. and Levinson, S. (1987). *Politeness: Some universals in language usage.* Cambridge University Press.

[Cahill et al., 2001] Cahill, L., Carroll, J., Evans, R., Paiva, D., Power, R., Scott, D., and van Deemter, K. (2001). From rags to riches: exploiting the potential of a flexible generation architecture. In *Meeting of the Association for Computational Linguistics*.

[Callison-Burch and Dredze, 2010] Callison-Burch, C. and Dredze, M. (2010). Creating speech and language data with amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 1–12, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Chen and Manning, 2014] Chen, D. and Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In Moschitti, A., Pang, B., and Daelemans, W., editors, *EMNLP*, pages 740–750. ACL.

[Chen and Mooney, 2008] Chen, D. L. and Mooney, R. J. (2008). Learning to sportscast: A test of grounded language acquisition. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 128–135, New York, NY, USA. ACM.

[Chen et al., 2018] Chen, M., Lampouras, G., and Vlachos, A. (2018). Sheffield at e2e: structured prediction approaches to end-to-end language generation. In *E2E NLG Challenge Description Systems.*

[Cho et al., 2014] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics.

[Dale et al., 1998] Dale, R., Eugenio, B. D., and Scott, D. (1998). Introduction to the special issue on natural language generation. *Computational-Linguistics, Volume 24, Number 3, September 1998.*

[Demberg and Moore, 2006] Demberg, V. and Moore, J. D. (2006). Information presentation in spoken dialogue systems. In *EACL*.

[Dethlefs et al., 2014] Dethlefs, N., Cuayáhuitl, H., Hastie, H., Rieser, V., and Lemon, O. (2014). Cluster-based prediction of user ratings for stylistic surface realisation. *EACL 2014*, page 702.

[Devillers et al., 2004] Devillers, L., Maynard, H., Rosset, S., Paroubek, P., McTait, K., Mostefa, D., Choukri, K., Charnay, L., Bousquet, C., Vigouroux, N., et al. (2004). The french media/evalda project: the evaluation of the understanding capability of spoken language dialogue systems. In *LREC*.

[Dewaele and Furnham, 1999] Dewaele, J.-M. and Furnham, A. (1999). Extraversion: the unloved variable in applied linguistic research. *Language Learning*, 49(3):509–544.

[Doddington, 2002] Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Dodge et al., 2015] Dodge, J., Gane, A., Zhang, X., Bordes, A., Chopra, S., Miller, A., Szlam, A., and Weston, J. (2015). Evaluating prerequisite qualities for learning end-to-end dialog systems. *arXiv preprint arXiv:1511.06931.*

[Dorr et al., 1998] Dorr, B. J., Habash, N., and Traum, D. R. (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.

[Dusek, 2017] Dusek, O. (2017). Novel methods for natural language generation in spoken dialogue systems. Technical report, Prague, Czech Republic.

[Dušek and Jurcicek, 2015] Dušek, O. and Jurcicek, F. (2015). Training a natural language generator from unaligned data. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 451–461, Beijing, China. Association for Computational Linguistics.

[Dušek and Jurcıcek, 2016] Dušek, O. and Jurcıcek, F. (2016). A context-aware natural language generator for dialogue systems. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue.*

[Dusek and Jurcícek, 2016] Dusek, O. and Jurcícek, F. (2016). Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *CoRR*, abs/1606.05491.

[Dušek et al., 2018] Dušek, O., Novikova, J., and Rieser, V. (2018). Findings of the e2e nlg challenge. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 322–328. Association for Computational Linguistics.

[Dusek et al., 2019] Dusek, O., Novikova, J., and Rieser, V. (2019). Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *CoRR*, abs/1901.07931.

[Elhadad and Robin, 1996] Elhadad, M. and Robin, J. (1996). An overview of surge: a reusable comprehensive syntactic realization component. In *INLG '96 Demonstrations and Posters*, pages 1–4, Brighton, UK. Eighth International Natural Language Generation Workshop.

[Fan et al., 2017] Fan, A., Grangier, D., and Auli, M. (2017). Controllable abstractive summarization. *CoRR*, abs/1711.05217.

[Farr et al., 1951] Farr, J. N., Jenkins, J. J., and Paterson, D. G. (1951). Simplification of Flesch Reading Ease Formula. *Journal of Applied Psychology*, 35(5):333–337.

[Ficler and Goldberg, 2017] Ficler, J. and Goldberg, Y. (2017). Controlling linguistic style aspects in neural language generation. *CoRR*, abs/1707.02633.

[Flanigan et al., 2014] Flanigan, J., Thomson, S., Carbonell, J., Dyer, C., and Smith, N. A. (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.

[Flesch, 1997] Flesch, R. F. (1997). How to write plain English: A book for lawyers and consumers. *Harper-Collins.*

[Fu et al., 2018] Fu, Z., Tan, X., Peng, N., Zhao, D., and Yan, R. (2018). Style transfer in text: Exploration and evaluation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 663–670.

[Gardent et al., 2017a] Gardent, C., Shimorina, A., Narayan, S., and Perez-

Beltrachini, L. (2017a). Creating Training Corpora for NLG Micro-Planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada.

[Gardent et al., 2017b] Gardent, C., Shimorina, A., Narayan, S., and Perez-Beltrachini, L. (2017b). Creating Training Corpora for NLG Micro-Planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada.

[Gašic et al., 2008] Gašic, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., Yu, K., and Young, S. (2008). Training and evaluation of the his-pomdp dialogue system in noise. *Proc. Ninth SIGdial, Columbus, OH.*

[Gasic et al., 2017] Gasic, M., Hakkani-Tur, D., and Celikyilmaz, A. (2017). Spoken language understanding and interaction: machine learning for human-like conversational systems. *Computer Speech and Language*, 46:249 – 251.

[Gatt and Krahmer, 2018] Gatt, A. and Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Int. Res.*, pages 65–170.

[Gehring et al., 2017] Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional Sequence to Sequence Learning. *ArXiv e-prints.*

[Gehrmann et al., 2018] Gehrmann, S., Dai, F., Elder, H., and Rush, A. (2018). End-to-end content and plan selection for data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 46–56, Tilburg University, The Netherlands. Association for Computational Linguistics.

[Ghosh et al., 2016] Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., and Heck, L. (2016). Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*.

[Gkatzia and Mahamood, 2015] Gkatzia, D. and Mahamood, S. (2015). A snapshot of nlg evaluation practices 2005 - 2014. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 57–60. Association for Computational Linguistics.

[Glorot and Bengio, 2010] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, page 249256.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[Gosling et al., 2003] Gosling, S. D., Rentfrow, P. J., and Swann, W. B. (2003). A very brief measure of the big five personality domains. *Journal of Research in Personality*, 37:504–528.

[H. Cheng and Mellish, 2001] H. Cheng, Massimo Poesio, R. H. and Mellish, C. (2001). Corpus-based np modifier generation. In *Proc. of the NAACL*.

[Hajdik et al., 2019] Hajdik, V., Buys, J., Goodman, M., and Bender, E. (2019). Neural text generation from rich semantic representations. In *To appear in Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT-19)*.

[Han et al., 2013] Han, L., Kashyap, A., Finin, T., Mayfield, J., and Weese, J. (2013). Umbc ebiquity-core: Semantic textual similarity systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, pages 44–52.

[Harrison et al., 2019] Harrison, V., Reed, L., Oraby, S., and Walker, M. (2019). Moving beyond blackbox neural language generation: Comparing methods for stylistic control. In *In submission.*

[Harrison and Walker, 2018] Harrison, V. and Walker, M. (2018). Neural generation of diverse questions using answer focus, contextual and linguistic features. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 296–306. Association for Computational Linguistics.

[Hastie and Belz, 2014] Hastie, H. and Belz, A. (2014). A comparative evaluation methodology for nlg in interactive systems. In Chair), N. C. C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

[Heilman et al., 2014] Heilman, M., Cahill, A., Madnani, N., Lopez, M., Mulholland, M., and Tetreault, J. (2014). Predicting Grammaticality on an Ordinal Scale. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 174–180.

[Herzig et al., 2017] Herzig, J., Shmueli-Scheuer, M., Sandbank, T., and Konopnicki, D. (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.

[Higashinaka et al., 2007a] Higashinaka, R., Walker, M., and Prasad, R. (2007a). Learning to generate naturalistic utterances using reviews in spoken dialogue systems. *ACM Transactions on Speech and Language Processing (TSLP).*

[Higashinaka et al., 2007b] Higashinaka, R., Walker, M. A., and Prasad, R. (2007b). An unsupervised method for learning generation dictionaries for spoken dialogue systems by mining user reviews. *ACM Transactions on Speech and Language Processing*, 4(4).

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Howcroft et al., 2013] Howcroft, D. M., Nakatsu, C., and White, M. (2013). Enhancing the expression of contrast in the sparky restaurant corpus. *ENLG 2013*, page 30.

[Hu et al., 2017] Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. (2017). Toward controlled generation of text. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.

[Inkpen and Hirst, 2004] Inkpen, D. Z. and Hirst, G. (2004). Near-synonym choice in natural language generation. In Nicolas Nicolov, Kalina Bontcheva, G. A. and Mitkov, R., editors, *Recent Advances in Natural Language Processing III.* John Benjamins Publishing Company.

[Isard et al., 2006] Isard, A., Brockmann, C., and Oberlander, J. (2006). Individuality and alignment in generated dialogues. In *Proceedings of the 4th International Natural Language Generation Conference (INLG)*, pages 22–29.

[Iyyer et al., 2018] Iyyer, M., Wieting, J., Gimpel, K., and Zettlemoyer, L. (2018). Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

[Jagfeld et al., 2018] Jagfeld, G., Jenne, S., and Vu, N. T. (2018). Sequence-to-sequence models for data-to-text natural language generation: Word- vs. character-based processing and output diversity. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 221–232, Tilburg University, The Netherlands. Association for Computational Linguistics.

[Johnson et al., 2016] Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2016). Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

[Jones and Galliers, 1996] Jones, K. S. and Galliers, J. R. (1996). *Evaluating Natural Language Processing Systems: An Analysis and Review.* Springer-Verlag, Berlin, Heidelberg.

[Juraska et al., 2018] Juraska, J., Karagiannis, P., Bowden, K., and Walker, M. (2018). A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. In *HLT-NAACL*.

[Juraska and Walker, 2018] Juraska, J. and Walker, M. (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.

[Karlgren, 2004] Karlgren, J. (2004). The wheres and whyfores for studying textual genre computationally.

[Klein et al., 2018] Klein, G., Kim, Y., Deng, Y., Nguyen, V., Senellart, J., and Rush, A. (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.

[Kobus et al., 2017] Kobus, C., Crego, J., and Senellart, J. (2017). Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378. INCOMA Ltd.

[Kondadadi et al., 2013] Kondadadi, R., Howald, B., and Schilder, F. (2013). A statistical nlg framework for aggregated planning and realization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1406–1415, Sofia, Bulgaria. Association for Computational Linguistics.

[Konstas and Lapata, 2013] Konstas, I. and Lapata, M. (2013). A global model for concept-to-text generation. *Journal of Artificial Intelligence Research*, 48:305–346.

[Lampouras and Vlachos, 2016] Lampouras, G. and Vlachos, A. (2016). Imitation learning for language generation from unaligned data. In Calzolari, N., Matsumoto, Y., and Prasad, R., editors, *COLING*, pages 1101–1112. ACL.

[Langkilde and Knight, 1998] Langkilde, I. and Knight, K. (1998). Generation that exploits corpus-based statistical knowledge. In *Proc. of COLING-ACL.*

[Langkilde-Geary, 2002] Langkilde-Geary, I. (2002). An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proc. of the INLG.*

[Lapata, 2003] Lapata, M. (2003). Probabilistic text structuring: Experiments with sentence ordering. In *Proc. of the ACL.*

[Lavie and Agarwal, 2007] Lavie, A. and Agarwal, A. (2007). Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Lavoie and Rambow, 1997] Lavoie, B. and Rambow, O. (1997). A fast and portable realizer for text generation systems. In *Proc. of the Third Conference on Applied Natural Language Processing, ANLP97*, pages 265–268.

[Lebret et al., 2016] Lebret, R., Grangier, D., and Auli, M. (2016). Neural text generation from structured data with application to the biography domain. In

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1203–1213. Association for Computational Linguistics.

[Lester and Porter, 1997] Lester, J. C. and Porter, B. W. (1997). Developing and Empirically Evaluating Robust Explanation Generators : The KNIGHT Experiments. *Computational Linguistcs*, 23(1):65–101.

[Lin, 2004] Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

[Luong et al., 2015] Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

[Mairesse, 2008] Mairesse, F. (2008). *Learning to Adapt in Dialogue Systems: Data-driven Models for Personality Recognition and Generation.* PhD thesis, University of Sheffield, United Kingdom.

[Mairesse et al., 2010] Mairesse, F., Gašić, M., Jurčíček, F., Keizer, S., Thomson, B., Yu, K., and Young, S. (2010). Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1552–1561, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Mairesse and Walker, 2007] Mairesse, F. and Walker, M. (2007). Personage: Personality generation for dialogue. In *In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL*, pages 496–503.

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

[Mairesse and Walker, 2011] Mairesse, F. and Walker, M. A. (2011). Controlling user perceptions of linguistic style: Trainable generation of personality traits. *Computational Linguistics*.

[Mairesse and Young, 2014] Mairesse, F. and Young, S. (2014). Stochastic language generation in dialogue using factored language models. *Computational Linguistics*, 40(4):763–799.

[Marcu, 1997] Marcu, D. (1997). From local to global coherence: a bottom-up approach to text planning. In *Proc. of the National Conference on Artificial Intelligence (AAAI'97)*.

[Mason and Watts, 2009] Mason, W. and Watts, D. J. (2009). Financial incentives and the "performance of crowds". In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '09, pages 77–85, New York, NY, USA. ACM.

[McAdams and Pals, 2006] McAdams, D. and Pals, J. (2006). A new Big Five: Fundamental principles for an integrative science of personality. *American Psychologist*, 61(3):204.

[Mehl et al., 2006] Mehl, M. R., Gosling, S. D., and Pennebaker, J. W. (2006). Personality in its natural habitat: Manifestations and implicit folk theories of personality in daily life. *Journal of Personality and Social Psychology*, 90:862–877.

[Mei et al., 2016] Mei, H., Bansal, M., and Walter, M. R. (2016). What to talk about and how? selective generation using lstms with coarse-to-fine alignment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 720–730, San Diego, California. Association for Computational Linguistics.

[Mel'cuk, 1988] Mel'cuk, I. A. (1988). Dependency syntax: Theory and practice. SUNY Press.

[Mellish and Dale, 1998] Mellish, C. and Dale, R. (1998). Evaluation in the context of natural language generation. *Computer Speech and Language*, 12(4):349–373.

[Meteer, 1990] Meteer, M. W. (1990). The "generation gap" the problem of expressibility in text planning. Technical report, Amherst, MA, USA.

[Mikolov et al., 2011] Mikolov, T., Kombrink, S., Burget, L., ernock, J., and Khudanpur, S. (2011). Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

[Moore and Paris, 1993] Moore, J. D. and Paris, C. L. (1993). Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4).

[Nayak et al., 2017] Nayak, N., Hakkani-Tur, D., Walker, M., and Heck, L. (2017).

To plan or not to plan? discourse planning in slot-value informed sequence to sequence models for language generation. In *Proc. of Interspeech 2017*.

[Novikova et al., 2017a] Novikova, J., Dušek, O., Cercas Curry, A., and Rieser, V. (2017a). Why we need new evaluation metrics for nlg. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252. Association for Computational Linguistics.

[Novikova et al., 2017b] Novikova, J., Dušek, O., and Rieser, V. (2017b). 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.

[Novikova et al., 2017c] Novikova, J., Dušek, O., and Rieser, V. (2017c). The E2E NLG shared task.

[Novikova et al., 2016] Novikova, J., Lemon, O., and Rieser, V. (2016). Crowdsourcing nlg data: Pictures elicit better data. *arXiv preprint arXiv:1608.00339*.

[Oberlander and Gill, 2004] Oberlander, J. and Gill, A. (2004). Individual differences and implicit language: personality, parts-of-speech, and pervasiveness. In *Proceedings of the 26th Annual Conference of the Cognitive Science Society*, pages 1035–1040.

[Oberlander and Gill, 2006] Oberlander, J. and Gill, A. J. (2006). Language with character: A stratified corpus comparison of individual differences in e-mail communication. *Discourse Processes*, 42:239–270.

[Oh and Rudnicky, 2002] Oh, A. H. and Rudnicky, A. I. (2002). Stochastic natural language generation for spoken dialog systems. *Computer Speech and Language: Special Issue on Spoken Language Generation*, 16(3-4):387–407.

[Oraby et al., 2019] Oraby, S., Harrison, V., Ebrahimi, A., and Walker, M. (2019). Curate and generate: A corpus and method for joint control of semantics and style in neural nlg. *(To appear at ACL 2019) CoRR*, abs/1906.01334.

[Oraby et al., 2016] Oraby, S., Harrison, V., Hernandez, E., Reed, L., Riloff, E., and Walker, M. (2016). Creating and characterizing a diverse corpus of sarcasm in dialogue. In *Proc. of the SIGDIAL 2016 Conference: The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue.*

[Oraby et al., 2017] Oraby, S., Homayon, S., and Walker, M. (2017). Harvesting creative templates for generating stylistically varied restaurant reviews. In *Proceedings of the Workshop on Stylistic Variation at EMNLP*, pages 28–36, Copenhagen, Denmark. Association for Computational Linguistics.

[Oraby et al., 2015] Oraby, S., Reed, L., Compton, R., Riloff, E., Walker, M., and Whittaker, S. (2015). And thats a fact: Distinguishing factual and emotional argumentation in online dialogue. In *NAACL HLT 2015 Workshop on Argument Mining*, page 116.

[Oraby et al., 2018a] Oraby, S., Reed, L., S., S. T., Tandon, S., and Walker, M. A. (2018a). Neural multivoice models for expressing novel personalities in dialog. In *19th Annual Conference of the International Speech Communication Association (Interspeech), Hyderabad, India, 2-6 September 2018.*, pages 3057–3061.

[Oraby et al., 2018b] Oraby, S., Reed, L., Tandon, S., TS, S., Lukin, S., and Walker, M. (2018b). 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 (SIGDIAL).*

[Oraby et al., 2018c] Oraby, S., Reed, L., Tandon, S., T.S., S., Lukin, S., and Walker, M. (2018c). Tnt-nlg, system 1: Using a statistical nlg to massively augment crowd-sourced data for neural generation. In *E2E NLG Challenge Description Systems.*

[Paiva and Evans, 2004] Paiva, D. S. and Evans, R. (2004). A framework for stylistically controlled generation. In Belz, A., Evans, R., and Piwek, P., editors, *Natural Language Generation, Third Internatonal Conference, INLG 2004*, number 3123 in LNAI, pages 120–129. Springer.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.*

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

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, page 15321543.

[Perez-Beltrachini et al., 2016] Perez-Beltrachini, L., SAYED, R., and Gardent, C. (2016). Building rdf content for data-to-text generation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics:*

Technical Papers, pages 1493–1502, Osaka, Japan. The COLING 2016 Organizing Committee.

[Polifroni et al., 1992] Polifroni, J., Hirschman, L., Seneff, S., and Zue, V. (1992). Experiments in evaluating interactive spoken language systems. In *Proc. of the DARPA Speech and NL Workshop*, pages 28–33.

[Rambow et al., 2001] Rambow, O., Rogati, M., and Walker, M. (2001). Evaluating a trainable sentence planner for a spoken dialogue travel system. In *Proc. of the Meeting of the Association for Computational Lingustics, ACL 2001.*

[Rao and Tetreault, 2018] Rao, S. and Tetreault, J. (2018). Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer.

[Ratnaparkhi, 2000] Ratnaparkhi, A. (2000). Trainable methods for surface natural language generation. In *Proc. of First North American ACL*, Seattle, USA.

[Ratnaparkhi, 2002] Ratnaparkhi, A. (2002). Trainable approaches to surface natural language generation and their application to conversational dialog systems. *Computer Speech and Language: Special Issue on Spoken Language Generation*, 16(3-4):435–455.

[Reed et al., 2019] Reed, L., Harrison, V., Oraby, S., and Walker, M. (2019). Something old something new: Generating novel dialogue utterances using source blending. In *In submission.*

[Reed et al., 2018] Reed, L., Oraby, S., and Walker, M. (2018). Can neural generators for dialogue learn sentence planning and discourse structuring? In *Proceed-*

ings of the 11th International Conference on Natural Language Generation, pages 284–295. Association for Computational Linguistics.

[Reiter, 2018] Reiter, E. (2018). A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

[Reiter and Belz, 2009] Reiter, E. and Belz, A. (2009). An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Comput. Linguist.*, 35(4):529–558.

[Reiter and Dale, 2000] Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press.

[Reiter et al., 2005] Reiter, E., Sripada, S., Hunter, J., Yu, J., and Davy, I. (2005). Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1):137 – 169. Connecting Language to the World.

[Rieser and Lemon, 2010] Rieser, V. and Lemon, O. (2010). Empirical methods in natural language generation. chapter Natural Language Generation As Planning Under Uncertainty for Spoken Dialogue Systems, pages 105–120. Springer-Verlag, Berlin, Heidelberg.

[Riloff, 1996] Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 1044–1049.

[Riloff and Phillips, 2004] Riloff, E. and Phillips, W. (2004). An introduction to the sundance and autoslog systems. Technical report, Technical Report UUCS-04-015, School of Computing, University of Utah.

[Rudnicky et al., 1999] Rudnicky, A., Thayer, E., Constantinides, P., Tchou, C., Shern, R., Lenzo, K., Xu, W., and Oh, A. (1999). Creating natural dialogs in the carnegie mellon communicator system. In *Eurospeech*, pages 1531–1534.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

[Sakaguchi et al., 2014] Sakaguchi, K., Post, M., and Durme, B. V. (2014). Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT at ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 1–11.

[Sauper and Barzilay, 2009] Sauper, C. and Barzilay, R. (2009). Automatically generating wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 208–216. Association for Computational Linguistics.

[Scott and de Souza, 1990] Scott, D. R. and de Souza, C. S. (1990). Getting the message across in RST-based text generation. In Dale, R., Mellish, C., and Zock, M., editors, *Current Research in Natural Language Generation*. Academic Press, London.

[Sennrich et al., 2016] Sennrich, R., Haddow, B., and Birch, A. (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 Com-*

putational Linguistics: Human Language Technologies, pages 35–40. Association for Computational Linguistics.

[Sharma et al., 2017] Sharma, S., He, J., Suleman, K., Schulz, H., and Bachman, P. (2017). Natural language generation in dialogue using lexicalized and delexicalized data.

[Shaw, 1998] Shaw, J. (1998). Clause aggregation using linguistic knowledge. In *Proc. of the 8th International Workshop on Natural Language Generation*, Niagara-on-the-Lake, Ontario.

[Shen et al., 2017] Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. S. (2017). Style transfer from non-parallel text by cross-alignment. In *NIPS*, pages 6833–6844.

[Siddharthan, 2004] Siddharthan, A. (2004). Syntactic simplification and text cohesion. Technical report, Cambridge, United Kingdom.

[Socher et al., 2013] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (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.

[Sordoni et al., 2015] Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., and Dolan, B. (2015). A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714*.

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):19291958.

[Stent, 2002] Stent, A. (2002). A conversation acts model for generating spoken dialogue contributions. *Computer Speech and Language: Special Issue on Spoken Language Generation.*

[Stent et al., 2005] Stent, A., Marge, M., and Singhai, M. (2005). Evaluating Evaluation Methods for Generation in the Presence of Variation. *Computational Linguistics and Intelligent Text Processing*, (c):341–351.

[Stent and Molina, 2009] Stent, A. and Molina, M. (2009). Evaluating automatic extraction of rules for sentence plan construction. In *Proc. of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 290–297.

[Stent et al., 2004] Stent, A., Prasad, R., and Walker, M. (2004). Trainable sentence planning for complex information presentation in spoken dialogue systems. In *Meeting of the Association for Computational Linguistics.*

[Stent et al., 2002] Stent, A., Walker, M., Whittaker, S., and Maloor, P. (2002). User-tailored generation for spoken dialogue: An experiment. In *ICSLP*.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

[Tandon et al., 2018] Tandon, S., T.S., S., Oraby, S., Reed, L., Lukin, S., and Walker, M. (2018). Tnt-nlg, system 2: Data repetition and meaning representation manipulation to improve neural generation. In *E2E NLG Challenge Description Systems*.

[Thorne, 1987] Thorne, A. (1987). The press of personality: A study of conversations between introverts and extraverts. *Journal of Personality and Social Psychology*, 53(4):718.

[Vaswani et al., 2018] Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A., Gouws, S., Jones, L., Kaiser, Ł., Kalchbrenner, N., Parmar, N., Sepassi, R., Shazeer, N., and Uszkoreit, J. (2018). Tensor2tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 193–199. Association for Machine Translation in the Americas.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

[Vauquois, 1968] Vauquois, B. (1968). A survey of formal grammars and algorithms for recognition and transformation in mechanical translation. In *IFIP Congress*.

[Vedantam et al., 2014] Vedantam, R., Zitnick, C. L., and Parikh, D. (2014). Cider: Consensus-based image description evaluation. *CoRR*, abs/1411.5726.

[Vinyals and Le, 2015] Vinyals, O. and Le, Q. (2015). A neural conversational model. *arXiv preprint arXiv:1506.05869.*

[Walker and Rambow, 2002] Walker, M. and Rambow, O., editors (2002). *Computer Speech and Language: Special Issue on Spoken Language Generation*, volume 16: 3-4. Academic Press.

[Walker et al., 2001] Walker, M., Rambow, O., and Rogati, M. (2001). Spot: A trainable sentence planner. In *Proc. of the North American Meeting of the Association for Computational Linguistics*.

[Walker et al., 2007] Walker, M., Stent, A., Mairesse, F., and Prasad, R. (2007). Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research (JAIR)*, 30:413–456.

[Walker et al., 2004] Walker, M. A., Whittaker, S., Stent, A., Maloor, P., Moore, J., Johnston, M., and Vasireddy, G. (2004). Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science*, 28(5):811–840.

[Wang et al., 2012] Wang, W. Y., Bohus, D., Kamar, E., and Horvitz, E. (2012). Crowdsourcing the acquisition of natural language corpora: Methods and observations. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 73–78.

[Wen et al., 2016] Wen, T.-H., Gasic, M., Mrksic, N., Rojas-Barahona, L. M., hao Su, P., Vandyke, D., and Young, S. J. (2016). Multi-domain neural network language generation for spoken dialogue systems. In *HLT-NAACL*.

[Wen et al., 2015] Wen, T.-H., Gasic, M., Mrksic, N., Su, P.-H., Vandyke, D., and Young, S. (2015). 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.

[Weston et al., 2015] Weston, J., Bordes, A., Chopra, S., Rush, A. M., van

Merriënboer, B., Joulin, A., and Mikolov, T. (2015). Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698.*

[Whittaker et al., 2002] Whittaker, S., Walker, M., and Moore, J. (2002). Fish or fowl: A Wizard of Oz evaluation of dialogue strategies in the restaurant domain. In *Language Resources and Evaluation Conference.*

[Wiseman et al., 2018] Wiseman, S., Shieber, S., and Rush, A. (2018). Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187, Brussels, Belgium. Association for Computational Linguistics.

[Wong and Mooney, 2007] Wong, Y. W. and Mooney, R. J. (2007). Generation by inverting a semantic parser that uses statistical machine translation. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT-07)*, pages 172–179, Rochester, NY.