Controlling Linguistic Style Aspects in Neural Language Generation

Jessica Ficler and **Yoav Goldberg**
Computer Science Department
Bar-Ilan University
Israel
{jessica.ficler, yoav.goldberg}@gmail.com

Abstract

Most work on neural natural language generation (NNLG) focus on controlling the content of the generated text. We experiment with controlling several stylistic aspects of the generated text, in addition to its content. The method is based on conditioned RNN language model, where the desired content as well as the stylistic parameters serve as conditioning contexts. We demonstrate the approach on the movie reviews domain and show that it is successful in generating coherent sentences corresponding to the required linguistic style and content.

1 Introduction

The same message (e.g. expressing a positive sentiment towards the plot of a movie) can be conveyed in different ways. It can be long or short, written in a professional or colloquial style, written in a personal or impersonal voice, and can make use of many adjectives or only few.

Consider for example the following to sentences:

(1) *"A genuinely unique, full-on sensory experience that treads its own path between narrative clarity and pure visual expression."*
(2) *"OMG... This movie actually made me cry a little bit because I laughed so hard at some parts lol."*

They are both of medium length, but the first appears to be written by a professional critic, and uses impersonal voice and many adjectives; while the second is written in a colloquial style, using a personal voice and few adjectives.

In a text generation system, it is desirable to have control over such stylistic aspects of the text: style variations are used to express the social meanings of a message, and controlling the style of a text is necessary for appropriately conveying a message in a way that is adequate to the social context (Biber and Conrad, 2009; Niederhoffer and Pennebaker, 2002). This work focuses on generating text while allowing control of its stylistic properties.

The recent introduction of recurrent neural language models and recurrent sequence-to-sequence architectures to NLP brought with it a surge of work on natural language generation. Most of these research efforts focus on controlling the *content* of the generated text (Lipton et al., 2015; Kiddon et al., 2016; Lebret et al., 2016; Kiddon et al., 2016; Tang et al., 2016; Radford et al., 2017), while a few model more stylistic aspects of the generated text such as the identity of the speaker in a dialog setting (Li et al., 2016); the politeness of the generated text or the text length in a machine-translation setting (Sennrich et al., 2016; Kikuchi et al., 2016); or the tense in generated movie reviews (Hu et al., 2017). Each of these works targets a single, focused stylistic aspect of the text. *Can we achieve finer-grained control over the generated outcome, controlling several stylistic aspects simultaneously?*

We explore a simple neural natural-language generation (NNLG) framework that allows for high-level control on the generated content (similar to previous work) as well as control over multiple stylistic properties of the generated text. We show that we can indeed achieve control over each of the individual properties.

As most recent efforts, our model (section 3) is based on a conditioned language model, in which the generated text is conditioned on a context vector.[1] In our case, the context vector encodes a set

[1] See (Hoang et al., 2016) for other conditioning models.

of desired properties that we want to be present in the generated text.[2] At training time, we work in a fully supervised setup, in which each sentence is labeled with a set of linguistic properties we want to condition on. These are encoded into the context vector, and the model is trained to generate the sentence based on them. At test time, we can set the values of the individual properties to get the desired response. As we show in section 6.3, the model generalizes fairly well, allowing the generation of text with property combinations that were not seen during training.

The main challenge we face is thus obtaining the needed annotations for training time. In section 4 we show how such annotations can be obtained from meta-data or using specialized text-based heuristics.

Recent work (Hu et al., 2017) tackles a similar goal to ours. They propose a novel generative model combining variational auto-encoders and holistic attribute discriminators, in order to achieve individual control on different aspects of the generated text. Their experiments condition on two aspects of the text (sentiment and tense), and train and evaluate on sentences of up to 16 words. In contrast, we propose a much simpler model and focus on its application in a realistic setting: we use all naturally occurring sentence lengths, and generate text according to two content-based parameters (sentiment score and topic) and four stylistic parameters (the length of the text, whether it is descriptive, whether it is written in a personal voice, and whether it is written in professional style). Our model is based on a well-established technology - conditioned language models that are based on Long Short-Term Memory (LSTM), which was proven as strong and effective sequence model.

We perform an extensive evaluation, and verify that the model indeed learns to associate the different parameters with the correct aspects of the text, and is in many cases able to generate sentences that correspond to the requested parameter values. We also show that conditioning on the given properties in a conditioned language model indeed achieve better perplexity scores compared to an unconditioned language model trained on the entire dataset, and also compared to unconditioned models that are trained on subsets of the data that correspond to a particular conditioning set. Finally, we show that the model is able to generalize, i.e., to generate sentences for combinations that were not observed in training.

2 Task Description and Definition

Our goal is to generate natural language text that conforms to a set of content-based and stylistic properties. The generated text should convey the information requested by the content properties, while conforming to the style requirements posed by the style properties.

For example in the movie reviews domain, `theme` is a content parameter indicating the topical aspect which the review refers to (i.e. the plot, the acting, and so on); and `descriptive` is a style parameter that indicates whether the review text uses many adjectives. The sentence *"A wholly original, well-acted, romantic comedy that's elevated by the modest talents of a lesser known cast."* corresponds to `theme:acting` and `descriptive:true`, as it includes many descriptions and refers to the acting, while the sentence *"In the end, there are some holes in the story, but it's an exciting and tender film."* corresponds to `theme:plot` and `descriptive:false`.

More formally, we assume a set of k parameters $\{p_1, \ldots, p_k\}$, each parameter p_i with a set of possible values $\{v_1, \ldots, v_{p_i}\}$. Then, given a specific assignment to these values our goal is to generate a text that is compatible with the parameters values. Table 1 lists the full set of parameters and values we consider in this work, all in the movie reviews domain. In section 4 we discuss in detail the different parameters and how we obtain their values for the texts in our reviews corpus.

To give a taste of the complete task, we provide two examples of possible value assignments and sentences corresponding to them:

Type	Parameter	Value (1)	Value (2)
Content	Theme	Acting	Other
Content	Sentiment	Positive	Negative
Style	Professional	True	False
Style	Personal	False	True
Style	Length	21-40 words	11-20 words
Style	Descriptive	False	True

Sentences for value set 1:

- "This movie is excellent, the actors aren't all over the place ,but the movie has a lot of fun, exploring the lesson in a way that they can hold their own lives."

[2] Another view is that of an encoder-decoder model, in which the encoder component encodes the set of desired properties.

	Parameter	Description	Source	Possible values	Examples
Style	Professional	Whether the review is written in the style of a professional critic or not	meta-data	False	“So glad to see this movie !!”
				True	“This is a breath of fresh air, it’s a welcome return to the franchise’s brand of satirical humor.”
	Personal	Whether the review describes subjective experience (written in personal voice) or not	content derived	False	“Very similar to the book.”
				True	“I could see the movie again, “The Kid With Me” is a very good film.”
	Length	Number of words	content derived	$\leq$ 10 words / 11-20 words / 21-40 words / $>$ 40 words	
	Descriptive	Whether the review is in descriptive style or not	content derived	True	“Such a hilarious and funny romantic comedy.”
				False	“A definite must see for fans of anime fans, pop culture references and animation with a good laugh too.”
Content	Sentiment	The score that the reviewer gave the movie	meta-data	Positive	“In other words: “The Four” is so much to keep you on the edge of your seat.”
				Neutral	“While the film doesn’t quite reach the level of sugar fluctuations, it’s beautifully animated.”
				Negative	“At its core ,it’s a very low-budget movie that just seems to be a bunch of fluff.”
	Theme	Whether the sentence’s content is about the *Plot*, *Acting*, *Production*, *Effects* or none of these (*Other*)	content derived	Plot	“The characters were great and the storyline had me laughing out loud at the beginning of the movie.”
				Acting	“The only saving grace is that the rest of the cast are all excellent and the pacing is absolutely flawless.”
				Production	“If you’re a Yorkshire fan, you won’t be disappointed, and the director’s magical.”
				Effects	“Only saving grace is the sound effects.”
				Other	“I’m afraid that the movie is aimed at kids and adults weren’t sure what to say about it.”

Table 1: Parameters and possible values in the movie-reviews domain.

- “It’s a realistic and deeply committed performance from the opening shot, the movie gives an excellent showcase for the final act, and the visuals are bold and daring.”

Sentences for value set 2:

- “My biggest gripe is that the whole movie is pretty absurd and I thought it was a little too predictable.”
- “The first half is pretty good and I was hoping for a few funny moments but not funny at all.”

3 Conditioned Language Model

Like in previous neural language-generation work (Lipton et al., 2015; Tang et al., 2016), our model is also a conditioned language model. In a regular language model (LM), each token w_t is conditioned on the previous tokens, and the probability of a sentence $w_1, ..., w_n$ is given by:

$$P(w_1, ..., w_n) = \Pi_{t=1}^{n} P(w_t | w_1, \dots w_{t-1}) \quad (1)$$

In a conditioned language model, we add an additional conditioning context, c:

$$P(w_1, ..., w_n | c) = \Pi_{t=1}^{n} P(w_t | w_1, \dots w_{t-1}, c) \quad (2)$$

Each token in the sentence is conditioned on the previous ones, as well the additional context c.

A conditioned language model can be implemented using an recurrent neural network language model (RNN-LM, (Mikolov et al., 2010)), where the context $\mathbf{c}$ is a vector that is concatenated to the input vector at each time step.

Conditioned language models were shown to be effective for natural language generation. We differ from previous work by the choice of conditioning contexts, and by conditioning on many parameters simultaneously.

In our case, the condition vector $\mathbf{c}$ encodes the desired textual properties. Each parameter value is associated with an embedding vector, and $\mathbf{c}$ is a concatenation of these embedding vectors. The vector $\mathbf{c}$ is fed into the RNN at each step, concatenated to the previous word in the sequence.

Technical Details We use an LSTM-based language model (Hochreiter and Schmidhuber, 1997), and encode the vocabulary using Byte Pair Encoding (BPE), which allows representation of an open vocabulary through a fixed-size vocabulary by splitting rare words into subword units, providing a convenient way of dealing with rare words. Further details regarding layer sizes, training regime, vocabulary size and so on are provided in the supplementary material.

4 Data-set Collection and Annotation

For training the model, we need a dataset of review texts, each annotated with a value assignment to each of the style and the content parameters. We obtain these values from two sources: (1) We derive it from meta-data associated with the review, when available. (2) We extract it from the review text using a heuristic. We use three kinds of heuristics: based on lists of content-words; based on the existence of certain function words; and based on the distribution on part-of-speech tags. These annotations may contain noise, and indeed some of our heuristics are not very tight. We demonstrate that we can achieve good performance despite the noise. Naturally, improving the heuristics is likely to results in improved performance.

Our reviews corpus is based on the Rotten-Tomatoes website.[3] We collected 1,002,625 movie reviews for 7,500 movies and split them into sentences. Each sentence is then annotated according to four style parameters (`professional`, `personal`, `descriptive` and `length`) and two content parameters (`sentiment` and `theme`). The meanings of these properties and how we obtain values for them are described below.

4.1 Annotations Based on Meta-data

Professional indicates whether the review is written in a professional (`true`) or a colloquial (`false`) style. We label sentences as `professional:true` if it is written by either (1) a reviewer that is a professional critic; (2) a reviewer that is marked as a "super-reviewer" on the RottenTomatoes website (a title given to reviewers who write high-quality reviews). Other sentences are labeled as `professional:false`.

Sentiment reflects the grade that was given by the review writer. Possible values for grade are: `positive`, `neutral`, `negative` or `none`. In audience reviews the movies are rated by the reviewer on a scale of 0 to 5 stars. In critic reviews, the score was taken from the original review (which is external to the rotten-tomatoes website). We normalized the critics scores to be on 0-5 scale. We then consider reviews with grade 0-2 as `negative`, 3 as `neutral` and 4-5 as `positive`. Cases where no score information was available are labeled as `none`.[4]

4.2 Annotations Derived from Text

Length We count the number of tokens in the sentence and associate each sentence to one of four bins: $\leq$10, 11-20, 21-40, $>$40.

Personal whether the sentence is written in a personal voice, indicating a subjective point of view (*"I thought it was a good movie."*, *"Just not my cup of tea."*) or not (*"Overall, it is definitely worth watching."*, *"The movie doesn't bring anything new."*), We label sentences that include the personal pronoun or possessive (*"I"*, *"my"*) as `personal:true` and others as `personal:false`.

Theme the aspect of the movie that the sentence refers to. The possible values are `plot`, `acting`, `production` and `effects`. We assign a category to a sentence using word lists. We went over the frequent words in the corpus, and looked for words that we believe are indicative of the different aspects (i.e., for `plot` this includes words such as *sciprt, story, subplots*. The complete word lists are available in the supplementary material). Each sentence was labeled with the category that has the most words in the sentence. Sentences that do not include any words from our lists are labeled as `other`.

Descriptive whether the sentence is descriptive (*"A warm and sweet, funny movie."*) or not (*"It's one of the worst movies of the year, but it's not a total waste of time."*), Our (somewhat simplistic) heuristic is based on the premise that descriptive texts make heavy use of adjectives. We labeled a sentence as `descriptive:true` if at least

[3] http://www.rottentomatoes.com

[4] Note that while the sentiment scores are assigned to a complete review, we associate them here with individual sentences. This is a deficiency in the heuristic, which may explain some of the failures observed in section 6.1.

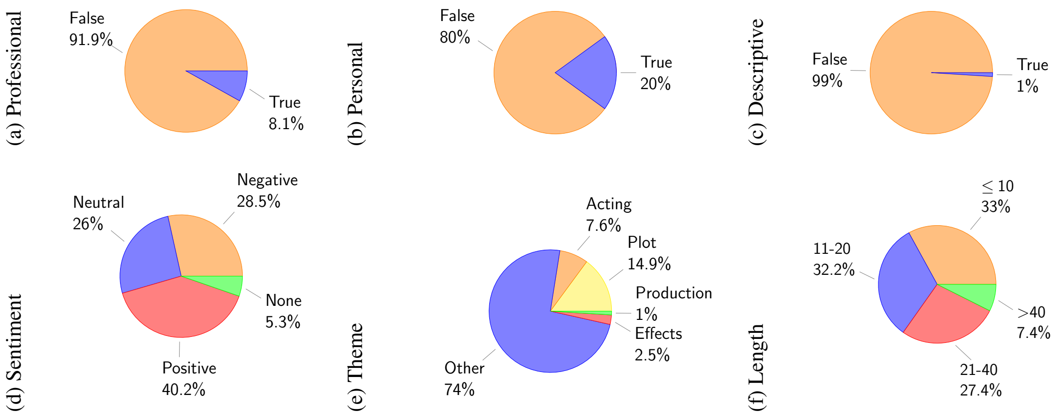

Figure 1: Movie reviews data-set statistics.

35% of its part-of-speech sequence tags are adjectives (JJ). All other sentences were considered as non-descriptive.

4.3 Dataset Statistics

Our final data-set includes 2,773,435 sentences where each sentence is labeled with the 6 parameters. We randomly divided the data-set to training (#2,769,138), development (#2,139) and test (#2,158) sets. Figure 1 shows the distribution of the different properties in the dataset.

5 Evaluating Language Model Quality

In our first set of experiments, we measure the quality of the conditioned language model in terms of test-set perplexity.

5.1 Conditioned vs. Unconditioned

Our model is a language model that is conditioned on various parameters. As a sanity check, we verify that knowing the parameters indeed helps in achieving better language modeling results. We compare the dev-set and test-set perplexities of our conditioned language model to an unconditioned (regular) language model trained on the same data. The results, summarized in the following table, show that knowing the correct parameter values indeed results in better perplexity.

	dev	test
Not-conditioned	25.8	24.4
Conditioned	**24.8**	**23.3**

Table 2: Conditioned and not-conditioned language model perplexities on the development and test sets.

5.2 Conditioned vs. Dedicated LMs

A second natural baseline to the conditioned LM is to train a separate unconditioned LM on a subset of the data. For example, if we are interested in generating sentences with the properties `personal:false`, `sentiment:pos`, `professional:false`, `theme:other` and `length:`$\leq$`10`, we will train a dedicated LM on just the sentences that fit these characteristics.

We hypothesize that the conditioned LM trained on all the data will be more effective than a dedicated LM, as it will be able to generalize across properties-combinations, and share data between the different settings. In this set of experiment, we verify this hypothesis.

For a set of parameters and values $\{p_1, p_2, \cdots p_n\}$, we train n sub-models where each sub-model m_i is trained on the subset of sentences that match parameters $\{p_1, p_2, \cdots p_i\}$. For example, given the set of parameters values as above, we train 5 sub-models: the first on data with `personal:false` only, the second on data with `persoal:false` and `sentiment:positive`, etc. As we add parameters, the size of the training set of the sub-model decreases.

For each dedicated sub-model, we measure its perplexity on the test-set sentences that match the criteria, and compare it to a conditioned LM with these criteria, and to an unconditioned language model. We do this for 4 different parameter-sets. Figure 2 presents the results.

The results indicate that when only few conditioning parameters are needed, and if the coverage of the parameter combination in the training set is large enough, the dedicated LM approach in-

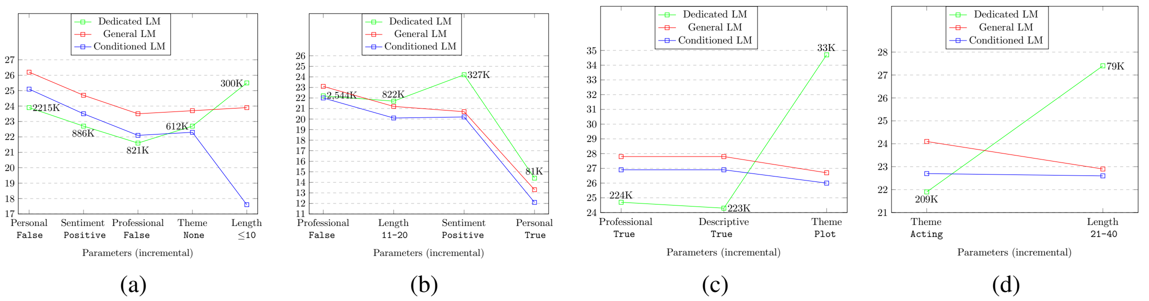

Figure 2: Perplexities of conditioned, unconditioned and dedicated language models for various parameter combinations. The numbers on the dedicated-model line indicates the number of sentences that the sub-model was trained on.

deed outperforms the conditioned LM. This is the case in the first three sub-models in 2a, and the first two sub-models in 2c. With few conditioning criteria, the dedicated LM approach is effective. However, it is not scalable. As we increase the number of conditioning factors, the amount of available training data to the dedicated model drops, and so does the modeling quality. In contrast, the conditioned model manages to generalize from sentences with different sets of properties, and is effective also with large number of conditioning factors. We thus conclude that for our use case, in which we need to condition on many different aspects of the generated sentence, the conditioned LM is far more suitable than the dedicated LM.

5.3 Conditioned vs. Flipped Conditioning

The previous experiments show that a conditioned model outperforms an unconditioned one. Here, we focus on the effect of the individual conditioning parameters. We compare the perplexity when using the correct conditioning values to the perplexity achieved when flipping the parameter value to an incorrect one. We do that for parameters that have opposing values: `personal`, `professional`, `sentiment` and `descriptive`. The following table summarizes the results:

Correct Value	23.3
Replacing Descriptive with non-Descriptive	27.2
Replacing Personal	27.5
Replacing Professional	25
Replacing Sentiment `Pos` with `Neg`	24.3

Table 3: Test-set perplexities when supplying the correct parameter values and when supplying the opposite values.

There is a substantial drop in quality (increase in perplexity) when flipping the parameter values. The drop is smallest for sentiment, and largest for descriptiveness and personal voice. We conclude that the model distinguishes descriptive text and personal voice better than it distinguishes sentiment and professional text.

6 Evaluating the Generated Sentences

In section 5.3 we verified the effectiveness of the conditioned model by showing that flipping a conditioning parameter value results in worse perplexity. However, we still need to verify that the model indeed associates each parameter with the correct behavior. In this set of experiments, we use the model to generate random sentences with different conditioning properties, and measure how well they match the requested behavior.

We generated 3,285 sentences according to the following protocol: for each property-combination attested in the development set, we generated 1,000 random sentences conditioned on these properties. We then sorted the generated sentences according to their probability, and chose the top $k = (c_f/m_f) * 100$ sentences, where c_f is the frequency of the property-combination in the dev set and m_f is the frequency of the most frequent property-combination in the dev set.

This process resulted in 3,285 high-scoring but diverse sentences, with properties that are distributed according to the properties distribution in the development set.

6.1 Capturing Individual Properties

Length We measure the average, minimum and maximum lengths, and deviation of the sentences that were generated for a requested length value. The following table summarizes the results:

Requested Length	Avg	Min	Max	Deviation$_{m=2}$
`<=10`	7.6	1	21	0.2 %
`11-20`	20.6	5	25	2.6 %
`21-40`	34	7	49	0.6 %

Table 4: Average, minimum and maximum lengths of the sentences generated according to the correspond length value; as well as deviation percentage with margin (m) of 2.

The average length fits the required range for each of the values and the percentage of sentences that exceed the limits with margin 2 is between 0.2% to 2.6%.

Descriptive We measure the percentage of sentences that are considered as descriptive (containing >35% adjectives) when requesting `descriptive:true`, and when requesting `descriptive:false`. When requesting descriptive text, **85.7%** of the generated sentences fit the descriptiveness criteria. When requesting non-descriptive text, **96%** of the generated sentences are non-descriptive according to our criteria.

Personal Voice We measure the percentage of sentences that are considered as personal voice (containing the pronouns *I* or *my*) when requesting `personal:true`, and when requesting `personal:false`. **100%** of the sentence for which we requested personal voice were indeed in personal voice. When requesting non-personal text, **99.85%** of the sentences are indeed non-personal.

Theme For each of the possible theme values, we compute the proportion of the sentences that were generated with the corresponding value. The confusion matrix in the following table

shows that the vast majority of sentences are generated according to the requested theme.

Requested value	% Plot	% Acting	% Prod	% Effects	% Other
Plot	98.7	0.8	0	0.2	0.3
Acting	2.5	95.3	0	0.6	1.6
Production	0	0	97.4	2.6	0
Effects	0	5.9	0	91.7	2.4
Other	0.04	0.03	0	0.03	99.9

Table 5: Percentage of generated sentences from each theme, when requesting a given theme value.

Professional The `professional` property of the generated sentences could not be evaluated automatically, and we thus performed manual evaluation using Mechanical Turk. We randomly created 1000 sentence-pairs where one is generated with `professional:true` and the other with `professional:false` (the rest of the property values were chosen randomly). For example in the following sentence-pair the first is generated with `professional:true` and the second with `professional:false`:

(t) *"This film has a certain sense of imagination and a sobering look at the clandestine indictment."*
(f) *"I know it's a little bit too long, but it's a great movie to watch !!!!"*

The annotators were asked to determine which of the sentences was written by a professional critic. Each of the pairs was annotated by 5 different annotators. When taking a majority vote among the annotators, they were able to tell apart the professional from non-professional sentences generated sentences in **72.1%** of the cases.

When examining the cases where the annotators failed to recognise the desired writing style, we saw that in a few cases the sentence that was generated for `professional:true` was indeed not professional enough (e.g. *"Looking forward to the trailer."*, and that in many cases, both sentences could indeed be considered as either professional or not, as in the following examples:

(t) *"This is a cute movie with some funny moments, and some of the jokes are funny and entertaining."*
(f) *"Absolutely amazing story of bravery and dedication."*

(t) *"A good film for those who have no idea what's going on, but it's a fun adventure."*
(f) *"An insult to the audience's intelligence."*

Sentiment To measure `sentiment` generation quality, we again perform manual annotations using Mechanical Turk. We randomly created 300 pairs of generated sentences for each of the following settings: `positive/negative`, `positive/neutral` and `negative/neutral`. The annotators were asked to mark which of the reviewers liked the movie more than the other. Each of the pairs was annotated by 5 different annotators and we choose by a majority vote. The annotators correctly identified **86.3%** of the sentence in the Positive/Negative case, **63%** of the sentences in the Positive/Neutral case, and **69.7%** of the sentences in the `negative`/`neutral` case.

Below are some examples for cases where the annotators failed to recognize the intended sentiment:

(Pos) *"It's a shame that this film is not as good as the previous film, but it still delivers."*
(Neg) *"The premise is great, the acting is not bad, but the special effects are so bad."*

(Pos) *"The story line is a bit predictable but it's a nice one, sweet and hilarious in its own right."*
(Neg) *"It's a welcome return to form an episode of Snow White, and it turns in a great way."*

6.2 Examples of Generated Sentences

All of the examples throughout the paper were generated by the conditioned LM. Additional examples are available in the supplementary material.

6.3 Generalization Ability

Finally, we test the ability of the model to generalize: can it generate sentences for parameter combinations it has not seen in training? To this end, we removed from the training set the 75,421 sentences which were labeled as `theme:plot` and `personal:true`, and re-trained a conditioned LM. The trained model did see 336,567 examples of `theme:plot` and 477,738 examples of `personal:true`, but has never seen examples where both conditions hold together. We then asked the trained model to generate sentences with these parameter values. **100%** of the generated sentences indeed contained personal pronouns, and **82.4%** of them fit the `theme:plot` criteria (in comparison, a conditioned model trained on *all* the training data managed to fit the `theme:plot` criteria in **97.8%** of the cases). Some generated sentence examples are:

"Some parts weren't as good as I thought it would be and the acting and script were amazing."

"I had a few laughs and the plot was great, but the movie was very predictable."

"I really liked the story and the performances were likable and the chemistry between the two leads is great."

"I've never been a fan of the story, but this movie is a great film that is a solid performance from Brie Larson and Jacob Tremblay.

7 Related Work

In neural-network based models for language generation, most work focus on content that need to be conveyed in the generated text. Similar to our modeling approach, (Lipton et al., 2015; Tang et al., 2016) generates reviews conditioned on parameters such as category, and numeric rating scores. Some work in neural generation for dialog (Wen et al., 2015; Dušek and Jurcicek, 2016b,a) condition on a dialog act ("request", "inform") and a set of key,value pairs of information to be conveyed ("price=low, food=italian, near=citycenter"). The conditioning context is encoded either similarly to our approach, or by encoding the desired information as a string and using sequence-to-seqeunce modeling with attention. Mei et al. (2016) condition the content on a set of key,value pairs using an encoder-decoder architecture with a coarse-to-fine attention mechanism. Kiddon et al. (2016) attempt to generate a recipe given a list of ingredients that should be mentioned in the text, tracking the ingredients that were already mentioned to avoid repetitions. Lebret et al. (2016) condition on structured information in Wikipedia infoboxes for generating textual biographies. [5] These work attempt to control the content of the generated text, but not its style.

In other works, the conditioning context correspond to a specific writer or a group of writers. In generation of conversational dialog, Li et al. (2016) condition the text on the speaker's identity. While the conditioning is meant for improving the factual consistency of the utterances (i.e., keeping track of age, gender, location), it can be considered as conditioning on stylistic factors (capturing personal style and dialect). A recent work that explicitly controls the style of the generated text was introduced by Sennrich et al. (2016) in the context of Machine Translation. Their model translates English to German with a feature that encodes whether the generated text (in German) should express politeness. All these works, with the exception of Sennrich et al condition on parameters that were extracted from meta-data or some database, while Sennrich et al heuristically extracts the politeness information from the training data. Our

[5] Recent work by Radford et al. (2017) trained an unconditioned LSTM language model on movie reviews, and found in a post-hoc analysis a single hidden-layer dimension that allows controling the sentiment of the generated reviews by fixing its value. While intriguuing, it is not a reliable method of deriving controllable generation models.

work is similar to the approach of Sennrich et al but extends it by departing from machine translation, conditioning on numerous stylistic aspects of the generated text, and incorporating both metadata and heuristically derived properties.

The work of Hu et al. (2017) features a VAE based method coupled with a discriminator network that tackles the same problem as ours: conditioning on multiple aspects of the generated text. The Variational component allows for easy sampling of examples from the resulting model, and the discriminator network directs the training process to associate the desired behavior with the conditioning parameters. Compared to our work, the VAE component is indeed a more elegant solution to generating a diverse set of sentences. However, the approach does not seem to be scalable: Hu et al. (2017) restrict themselves to sentences of up to length 16, and only two conditioning aspects (sentiment and tense). We demonstrate that our conditioned LSTM-LM appraoch easily scales to naturally-occuring sentence lengths, and allows control of 6 individual aspects of the generated text, without requiring a dedicated discriminator network. The incorporation of a variational component is an interesting avenue for future work.

In Pre-neural Text Generation The incorporation of stylistic aspects was discussed from very early on (McDonald and Pustejovsky, 1985). Some works tackling stylistic control of text produced in a rule-based generation system include the works of Power et al. (2003); Reiter and Williams (2010); Hovy (1987); Bateman and Paris (1989) (see (Mairesse and Walker, 2011) for a comprehensive review). Among these, the work of Power et al. (2003), like ours, allows the user to control various stylistic aspects of the generated text. This works by introducing soft and hard constraints in a rule-based system. The work of Mairesse and Walker (2011) introduce statistics into the stylistic generation process, resulting in a system that allows a user to specify 5 personality traits that influence the generated language.

More recent statistical generation works tackling style include Xu et al. (2012) who attempt to paraphrase text into a different style. They learn to paraphrase text in Shakespeare's style to modern English using MT techniques, relying on the modern translations of William Shakespeare plays. Abu Sheikha and Inkpen (2011) generate texts with different formality levels by using lists of formal and informal words.

Finally, our work relies on heuristically extracting stylistic properties from text. Computational modeling of stylistic properties has been the focus of several lines of study, i.e. (Pavlick and Tetreault, 2016; Yang and Nenkova, 2014; Pavlick and Nenkova, 2015). Such methods are natural companions for our conditioned generation approach.

8 Conclusions

We proposed a framework for NNLG allowing for relatively fine-grained control on different stylistic aspects of the generated sentence, and demonstrated its effectiveness with an initial case study in the movie-reviews domain. A remaining challenge is providing finer-grained control on the generated *content* (allowing the user to specify either almost complete sentences or a set of structured facts) while still allowing the model to control the style of the generated sentence.

Acknowledgements The research was supported by the Israeli Science Foundation (grant number 1555/15) and the German Research Foundation via the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1).

References

Fadi Abu Sheikha and Diana Inkpen. 2011. Generation of formal and informal sentences. In *Proceedings of the 13th European Workshop on Natural Language Generation*. Association for Computational Linguistics, Nancy, France, pages 187–193. http://www.aclweb.org/anthology/W11-2826.

John A Bateman and Cecile Paris. 1989. Phrasing a text in terms the user can understand. In *IJCAI*. pages 1511–1517.

Douglas Biber and Susan Conrad. 2009. *Register, genre, and style*. Cambridge University Press.

Ondřej Dušek and Filip Jurcicek. 2016a. A context-aware natural language generator for dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, Los Angeles, pages 185–190. http://www.aclweb.org/anthology/W16-3622.

Ondřej Dušek and Filip Jurcicek. 2016b. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume*

2: Short Papers). Association for Computational Linguistics, Berlin, Germany, pages 45–51. http://anthology.aclweb.org/P16-2008.

Cong Duy Vu Hoang, Gholamreza Haffari, and Trevor Cohn. 2016. Incorporating side information into recurrent neural network language models. In *Proceedings of NAACL-HLT*. pages 1250–1255.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. MIT Press, volume 9, pages 1735–1780.

Eduard Hovy. 1987. Generating natural language under pragmatic constraints. Journal of Pragmatics, volume 11, pages 689–719.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Controllable text generation. In *Proc. of ICML*.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 329–339. https://aclweb.org/anthology/D16-1032.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1328–1338. https://aclweb.org/anthology/D16-1140.

Rémi Lebret, David Grangier, and Michael Auli. 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*. Association for Computational Linguistics, Austin, Texas, pages 1203–1213. https://aclweb.org/anthology/D16-1128.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 994–1003. http://www.aclweb.org/anthology/P16-1094.

Zachary C Lipton, Sharad Vikram, and Julian McAuley. 2015. Capturing meaning in product reviews with character-level generative text models. arXiv preprint arXiv:1511.03683.

François Mairesse and Marilyn A Walker. 2011. Controlling user perceptions of linguistic style: Trainable generation of personality traits. MIT Press, volume 37, pages 455–488.

David D McDonald and James D Pustejovsky. 1985. A computational theory of prose style for natural language generation. In *Proceedings of the second conference on European chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 187–193.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 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*. Association for Computational Linguistics, San Diego, California, pages 720–730. http://www.aclweb.org/anthology/N16-1086.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*. volume 2, page 3.

Kate G Niederhoffer and James W Pennebaker. 2002. Linguistic style matching in social interaction. Journal of Language and Social Psychology, volume 21, pages 337–360.

Ellie Pavlick and Ani Nenkova. 2015. Inducing lexical style properties for paraphrase and genre differentiation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, pages 218–224. http://www.aclweb.org/anthology/N15-1023.

Ellie Pavlick and Joel Tetreault. 2016. An empirical analysis of formality in online communication. Transactions of the Association for Computational Linguistics, volume 4, pages 61–74.

Richard Power, Donia Scott, and Nadjet Bouayad-Agha. 2003. Generating texts with style. In *Proc. of CiCLING*. Springer, pages 444–452.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. arXiv preprint arXiv:1704.01444.

Ehud Reiter and Sandra Williams. 2010. Generating texts in different styles. In *The Structure of Style*, Springer, pages 59–75.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of NAACL-HLT*. pages 35–40.

Jian Tang, Yifan Yang, Sam Carton, Ming Zhang, and Qiaozhu Mei. 2016. Context-aware natural language generation with recurrent neural networks. arXiv preprint arXiv:1611.09900.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young.

2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1711–1721. http://aclweb.org/anthology/D15-1199.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of COLING 2012*. The COLING 2012 Organizing Committee, Mumbai, India, pages 2899–2914. http://www.aclweb.org/anthology/C12-1177.

Yinfei Yang and Ani Nenkova. 2014. Detecting information-dense texts in multiple news domains. In *AAAI*. pages 1650–1656.