Controllable Abstractive Summarization

Angela Fan, David Grangier, Michael Auli
Facebook AI Research
Menlo Park, CA

Abstract

Current models for document summarization ignore user preferences such as the desired length, style or entities that the user has a preference for. We present a neural summarization model that enables users to specify such high level attributes in order to control the shape of the final summaries to better suit their needs. With user input, we show that our system can produce high quality summaries that are true to user preference. Without user input, we can set the control variables automatically and outperform comparable state of the art summarization systems despite the relative simplicity of our model.

1 Introduction

Automatic summarization condenses a document into a short paragraph or a single sentence while retaining the core information. Broadly, summarization algorithms are either extractive or abstractive. Extractive algorithms form summaries by pasting relevant portions of the input together, while abstractive summarization algorithms may generate new text that is not present in the initial document (Das and Martins, 2007; Nenkova et al., 2011).

This work focuses on abstractive summarization and, in contrast to previous work, describes a mechanism that enables the reader to control important aspects of the generated summary. The reader can select the desired *length* of the summary depending on how detailed they would like the summary to be. The reader can also require the text to focus on *entities* they have a particular interest in. Finally, we let the reader choose the *style* of the summary based on their favorite source of information, e.g., the writing style of a particular news source.

Our work builds on sequence-to-sequence models (Sutskever et al., 2014; Bahdanau et al., 2015), which have been extensively applied to the task of abstractive summarization (Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016; See et al., 2017; Paulus et al., 2017). These models are neural language models conditioned on an input document. The encoder module builds a representation of the input document and the decoder generates a summary by attending to the the source representation (Bahdanau et al., 2015). Recent summarization models build upon pointer networks (Nallapati et al., 2016; See et al., 2017; Paulus et al., 2017) and have a few main architectural differences. For instance, (See et al., 2017) pairs attention with a coverage mechanism to avoid repetition and (Paulus et al., 2017) relies on intra-decoder attention to enable generating coherent multi-sentence summaries.

We introduce a controllable summarization model that provides a mechanism for users to specify high level attributes such as length, style, or entities of interest. This enables personalized generation, hence fully leveraging the fact that automatic summaries are generated at the reader's request. We show that (1) our generated summaries follow the specified preferences and (2) these control variables guide the learning process, which improves generation even when the control variables are set automatically during inference. Our comparison with other models from the literature on the standard CNN/DailyMail benchmark (Nallapati et al., 2016) highlights the advantage of our approach. On both the entity-anonymized (+0.38 F1-ROUGE1) and full text versions (+0.22 F1-ROUGE1) of the dataset, we outperform previous pointer-based models trained with maximum likelihood despite the relative simplicity of our model.

The remainder of the paper is organized as follows. Section 2 presents our approach to controlled summarization. Section 3 describes related prior research. Section 4 shows our experimental findings. Finally, Section 6 presents our conclusions.

2 Controllable Summarization

In this section, we present our controllable summarization model and introduce several control variables that the user can modify.

2.1 Convolutional Sequence-to-Sequence

Our approach builds upon the convolutional encoder-decoder model from (Gehring et al., 2017). This model combines high quality generation with computationally efficient inference, as shown on translation and sentence compression benchmarks.

The encoder and decoder are based on convolutional networks (LeCun et al., 1990). Both are composed of several layers which start with a word embedding layer followed by alternating convolutions and non-linear Gated Linear Units, GLU (Dauphin et al., 2017). The decoder is connected to the encoder through an attention module (Bahdanau et al., 2015) after each GLU. The attention performs a weighted sum of the encoder outputs. The weights are predicted from the current decoder states, allowing the decoder to emphasize the parts of the input document which are the most relevant for generating the next token. In contrast to previous work, we use multi-hop attention, such that the attention is added at each layer of the decoder.

In addition to attending over encoder states (Bahdanau et al., 2015), we also use intra-attention in the decoder to enable the model to refer back to previously generated words at any time scale. The mechanism allows the decoder to keep track of its progress and dissuades the decoder from generating repeated information (Vaswani et al., 2017; Paulus et al., 2017). To combine encoder and decoder attention, we alternate between each type of attention at every layer.

Previous work used pointer networks to copy proper nouns and other entities from the input (Nallapati et al., 2016) which introduces additional complexity to the model. Instead, we rely on sub-word tokenization and weight sharing. This choice is simpler but also very effective as we demonstrate in our experiments. We use byte-pair-encoding (BPE) to tokenize the data which has been shown to enable copying of proper nouns in translation (Sennrich et al., 2016). We share the representation of the tokens in the encoder and decoder embeddings as well as in the last layer of the decoder.

2.2 Length-Constrained Summarization

Summarization allows a reader with a limited time budget to quickly access the essence of a document. Controlling the length of a summary enables reading with different time constraints. For example, as the users time budget increases, we may summarize an article as a five-word headline, a single sentence, or using multiple sentences, each providing more and more detail.

Specifically, we enable the user to control the length of a summary as follows: First, we assign the training summaries to a set of discrete buckets, each representing a size range that does not overlap with other buckets. We choose the ranges so that buckets contain roughly an equal number of documents. We then expand the input vocabulary with special types to indicate the length bin of a document. This allows us to condition generation upon the discrete length variable. At training time, we prepend the input of our summarizer with the marker that indicates the length of the ground-truth summary.

At test time, we control the length of the generated text by prepending a particular length marker token. Our experiments (Section 5.2) provide both quantitative and qualitative evidence that the model can effectively use this variable: we show that the output length is easily controlled by changing the length marker and that supplying ground truth markers drastically improves summary quality.

2.3 Entity-Centric Summarization

The reader might be interested in a document to learn about specific entities (e.g. people, locations, etc.). For instance, a sports fan who is reading about a recent game may always want to read about their favorite player. Therefore, it would be useful to enable the reader to specify which entities the automatic summary should include.

We enable entity-centric summaries as follows: First, we anonymize entities by replacing all occurrences of a particular entity within a document by the same token. For training data, we also anonymize the corresponding reference summary. This process is repeated for all entities: for each (document, summary) pair, we associate each entity with a token from the set $(@entity0, \ldots, @entityN)$. This abstracts away from the surface form of an entity. This process allows scaling to many entity types and enables generalizing to unseen entities at test time.

After anonymization, we express that an entity should be present in the generated summary by prepending the entity token (@entity0,...,@entityN) to the input. We also append a separator token to keep the entity marker and the article separate. For example, prepending @entity3 expresses that the model should generate a summary where @entity3 is present. In effect, this instructs the model to focus attention on sentences that mention the marked entities.

At training time, we prepend each article with markers referring to a randomly selected entity from the ground-truth summary and at inference time we specify an entity marker that we wish the summary to contain. Our experiments (Section 5.2) evaluate the effect of prepending different entity markers to the input. We show that higher accuracy can be achieved when we specify entities occurring in the first few sentences of a document or if we supply markers taken from the reference summary to illustrate specific user preferences.

2.4 Source-Specific Summarization

Newspapers, magazines, encyclopedia, etc. have specific writing style guidelines to promote a uniform experience for readers. Readers become accustomed to the styles of their favorite sources. Therefore, we would like to enable users to specify a preferred source style for a summary.

Similar to length and entity control, we introduce special marker tokens (@genSource0, ..., @genSourceN) to express source desiderata. At training time, we indicate the source of the reference summary by prepending the corresponding marker to the input of the model. At inference time, we can control the style of generated summary by prepending a particular marker to the input of the model.

Our experiments (see Section 4) evaluate whether providing the true source-style produces summaries that are closer to the reference summary. We additionally provide examples of distinct summaries that result from different source-style conditioning.

3 Related Work

This section describes prior work related to summarization and controllable text generation.

3.1 Sequence-to-Sequence Models for Summarization

Automatic summarization has been an active field of research for close to 60 years (Luhn, 1958). Efforts in both extractive and abstractive methods have followed advances in the field of natural language processing, pattern recognition, and machine learning (Nenkova et al., 2011). Recently, sequence-to-sequence neural networks (Sutskever et al., 2014) have been applied to abstractive summarization (Nallapati et al., 2016; See et al., 2017; Paulus et al., 2017) following their success in both machine translation (Bahdanau et al., 2015; Luong et al., 2015b), parsing (Luong et al., 2015a) and image captioning (Vinyals et al., 2015b).

Research in abstractive summarization with sequence-to-sequence models focuses on neural architectures (Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016; See et al., 2017; Paulus et al., 2017) and learning objectives (Paulus et al., 2017). Summarization models have benefited from architectural advances in machine translation and related fields. For instance, attention mechanisms (Bahdanau et al., 2015) enable generation to focus on a targeted part of the source document. Pointer mechanisms (Vinyals et al., 2015a) have been useful for abstractive summarization where copying entities and other rare words from the input is highly advantageous (See et al., 2017; Paulus et al., 2017).

Summarization also has distinct challenges. For instance, the generation of multi-sentence summaries differs from individual sentence translation: when doing left-to-right decoding, the decoder needs to be aware of its previous generation at a larger time scale, otherwise the network tends to produce repeated text. To address this impediment, (See et al., 2017) introduce coverage modeling, (Paulus et al., 2017) propose intra-decoder attention, and (Suzuki and Nagata, 2017) equip the decoder with an estimator of unigram frequency.

Regarding learning objectives, (Paulus et al., 2017) investigates the potential improvement from replacing maximum likelihood training with reinforcement learning to optimize ROUGE, the most common automatic metric to assess summarization. They find that a combination of both losses to perform best according to human evaluations, as training with reinforcement learning alone tends to produce non-grammatical text.

Our model builds upon this previous work. Following (Gehring et al., 2017), we rely on convolutional networks, in contrast to previous work using recurrent networks (Nallapati et al., 2016; See et al., 2017; Paulus et al., 2017). The convolutional networks enable faster training time and better batching. Like (Paulus et al., 2017), we rely on intra-attention for generating multi-sentence text. We introduce multi-hop intra-attention inspired by multi-hop source attention from (Gehring et al., 2017). This allows more complex attention patterns and improves accuracy. Like (Paulus et al., 2017), we share the word embeddings in the encoder and decoder lookup tables with the embeddings from the output layer of the decoder. This facilitates better copying of source entities. Contrary to (Paulus et al., 2017; See et al., 2017; Nallapati et al., 2016), we rely on Byte-Pair Encoding (BPE) (Sennrich et al., 2016) to improve the model copy mechanism instead of using an additional pointer mechanism.

Contrary to (Paulus et al., 2017), we did not explore training objectives and our training procedure aims at maximizing the likelihood of the training summaries given the source document. Our model is amenable to training with reinforcement learning, but this aspect is largely orthogonal to the main goal of our work, i.e. controllable summarization.

3.2 Controllable Text Generation

Text generation is an established research area (McKeown, 1992). The field follows recent advances in generative models, such as the introduction of variational auto-encoders (Kingma and Welling, 2013) and adversarial networks (Goodfellow et al., 2014). This is exemplified by work focusing on natural language generation such as (Bowman et al., 2016; Yu et al., 2017; Zhao et al., 2017; Rajeswar et al., 2017).

Building upon unconditioned generation, controllable generation is an emerging research field. Research in computer vision includes style transfer (Gatys et al., 2015) or controllable image generation (Lample et al., 2017). Text generation work focuses on controlling tense or sentiment with variational auto-encoders (Hu et al., 2017). (Shen et al., 2017) relies on adversarial training for controlling sentence sentiment. (Ficler and Goldberg, 2017) proposes on conditional language model to generate text with stylistic requirements. (Filippova, 2017) proposes controlling length for generating answers in a question answering task.

Set sizes	
train set	287,227 documents
valid set	13,368 documents
test set	11,490 documents
Average Length of Train	
document	384 tokens
summary	51 tokens
Vocabulary	
BPE Vocabulary	30k types

Table 1: CNN/Daily Mail Statistics

Our work relies on conditional language modeling and does not leverage adversarial training or latent variable models such as variational autoencoders. Our motivation is simplicity. While latent variable models are popular for the generation of continuous outputs such as images, (conditional) language models are flexible enough to capture the multimodal nature of the data. We leave the assessment of how additional latent variables might improve upon our results to future work.

4 Experimental Setup

This section introduces the data we evaluate on, the architecture of our model and our evaluation procedure.

4.1 Dataset

We use the CNN/Dailymail dataset (Hermann et al., 2015; Nallapati et al., 2016) which consists of online news articles along with multi-sentence summaries. The statistics of the dataset are reported in Table 1 after limiting the length of the train documents to 400, as suggested by (See et al., 2017). We evaluate on two versions of this dataset, the entity anonymized version (Hermann et al., 2015; Nallapati et al., 2016; Paulus et al., 2017) and the full text version (See et al., 2017)[1]. We use BPE with 30K types (Sennrich et al., 2016). For models not trained with BPE, the

[1] Downloaded from https://cs.nyu.edu/~kcho/DMQA/

source and target vocabulary consists of all words appearing at least 20 times, creating a source vocabulary size of 47,174 and a target vocabulary size of 21,214.

4.2 Architecture, Training, and Generation

We implement our models in Torch (Collobert et al., 2011) and on top of the fairseq library [2]. Our model architecture consists of 8 convolutional encoder and decoder layers, each with kernel width 3. We use 512 hidden units for the encoder and decoder side, and embedding size 256. We add dropout 0.2 to the convolutional and fully connected layers.

We train our models following (Gehring et al., 2017), using Nesterov's accelerated gradient method (Sutskever et al., 2013) with gradient clipping 0.1 (Pascanu et al., 2013), momentum 0.99, and learning rate 0.2. We reduce the learning rate by an order of magnitude when the validation perplexity ceases to improve, and terminate training when the learning rate drops below 10^{-5}. Summaries are generated using beam search with beam of size 5. To avoid repetition, we prevent the decoder from generating the same trigram more than once during test, following (Paulus et al., 2017).

4.3 Evaluation

We evaluate using the standard ROUGE metric (Lin, 2004) and report the F1 scores for ROUGE-1, ROUGE-2, and ROUGE-L. We compare to existing abstractive baselines (Nallapati et al., 2016; See et al., 2017; Paulus et al., 2017) and report results on the Lead-3 extraction baseline which simply selects the first three sentences of the input article as its summary.

5 Results

We first evaluate the design choices of our convolutional summarization model and then the impact of manipulating the individual control variables. Finally, we show that the control variables are generally beneficial for model performance. To that end, we quantify the ROUGE score when setting the control variables automatically at test time.

5.1 Convolutional Summarization

Table 2 details the effect of our design choices on top of Gehring et al. (2017). First, we add a constraint to avoid repeated trigrams at generation time which improves F1-ROUGE1 by +2.86. Next, we add intra-attention to enable the model to examine past generations over long distances. This improves the accuracy obtained with the trigram constraint by a further 0.51 F1-ROUGE1. The modest improvement is likely because the two features address a similar problem: the trigram constraint avoids repeated generations through constraining inference whereas intra-attention addresses the same problem through a change in the actual model. Finally, we switch data tokenization to BPE instead of a word-based vocabulary which gives another +0.79 F1-ROUGE1. The BPE vocabulary improves the ability to copy proper nouns and rare inflections, both of which are difficult to model in word-based vocabularies. Our improvements with BPE are comparable to results on machine translation (Sennrich et al., 2016).

Model	**ROUGE**		
	1	**2**	**L**
fairseq	33.32	12.64	30.57
+ trigram decoding	36.18	14.10	33.18
+ intra-attention	36.69	14.28	33.47
+ BPE	37.48	15.12	34.16

Table 2: Performance without control variables. Each row shows results of adding another feature to the previous set of features (e.g. row 3 depicts fairseq + trigram decoding + intra-attention).

5.2 Controllable Summarization

Model	**ROUGE**		
	1	**2**	**L**
baseline, no control	37.48	15.12	34.16
Length constraint	39.16	15.54	35.94
Entity centric	37.74	15.15	34.54
Source specific	37.68	15.16	34.40
Length+Entity+Source	39.61	15.83	36.48

Table 3: Summarization when setting the control variables to the ground truth, as if simulating true user preferences

Our summarizer allows the user to control the length of the generated summary, the entities on which it focuses on, as well as the source style it imitates (see Section 2). First, we evaluate the effect of providing the true reference variables at decoding time. This simulates a user which expresses preferences through specifying values of

[2] https://github.com/facebookresearch/fairseq

Model		ROUGE-1	ROUGE-2	ROUGE-L
Lead-3	(Nallapati et al., 2017)	39.2	15.7	35.5
ML	words-lvt2k-temp-att (Nallapati et al., 2016)	35.46	13.30	32.65
	ML, no intra-attention (Paulus et al., 2017)	37.86	14.69	34.99
	ML, with intra-attention (Paulus et al., 2017)	38.30	14.81	**35.49**
	Baseline without control variables (ours)	37.48	15.12	34.16
	Controlled summarization with fixed values (ours)	**38.68**	**15.40**	35.47
RL	ML+RL with intra-attention (Paulus et al., 2017)	**39.87**	**15.82**	**36.90**

Table 4: Performance of fixed control variables on entity-anonymized text. Even with a fixed setting, the controlled summarization model improves ROUGE compared to other models trained with MLE.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Lead-3 (See et al., 2017)	40.34	17.70	36.57
Pointer-generator + coverage (See et al., 2017)	39.53	17.28	36.38
Baseline without control variables (ours)	38.23	16.68	34.77
Controlled summarization with fixed values (ours)	**39.75**	**17.29**	**36.54**

Table 5: Summarization with fixed control variables on original text. Even with a fixed setting, the controlled summarization model improves ROUGE.

the control variables. Second, we evaluate the effect of providing non-reference control variables.

Table 3 reports the result of these experiments for each variable as well as their combined effect. All control variables improve the quality of the generated summary. However, length control is the most impactful variable, followed by entity control and source style. Furthermore, the advantages of each control variable can be combined to produce an even stronger summary: we obtain +2.2 F1-ROUGE1 when combining all three control variables. Next, we discuss each variable in detail.

Length control improves accuracy by 1.68 F1-ROUGE1 (Table 3). This improvement is due to two effects: (1) predicting a summary whose length is very different from the reference strongly impacts F1-ROUGE, and (2) the baseline model struggles in predicting the correct length. The latter is because there is large uncertainty in how long a summary should actually be. This is to the extent that even humans find it very difficult to predict the correct length.

Figure 1 demonstrates that the model respects the specified length constraint marker by generating longer summaries when instructed to do so. The figure reports the average summary length when decoding all examples in the test set using each of the 10 possible length marker tokens. Table 7 demonstrates the effect of the length marker on a specific example.

	Baseline	Entity-centric
Lead-3	15.28	**57.74**
Full input	7.64	**30.28**

Table 6: Percentage of entity appearance in decoded summary, evaluated on 100 documents. Entities are gathered from either Lead-3 or the full input document.

Entity control has less impact on ROUGE compared to length control at +0.26 vs. +1.68 F1-ROUGE1 (Table 3). The lower effect is an amalgamation of many possible factors. First, the baseline system already includes many correct entities in the summary. Second, the model broadly learns from the raw text where the important article information is, and thus the entity marker is useful mostly in cases where it provides additional signal. For example, the entity can be a useful indicator that the summary content may occur in a specific portion of the article that mentions this entity.

Table 6 provides an analysis of this on 100 test documents. For each entity appearing in Lead-3 and in the full source, we decode by requiring to focus on that entity, and measure how often the entity-centric model generates a summary that includes the requested entity. In both settings, the model mentions the requested entity more than the baseline. In the case of Lead-3 entities, the model

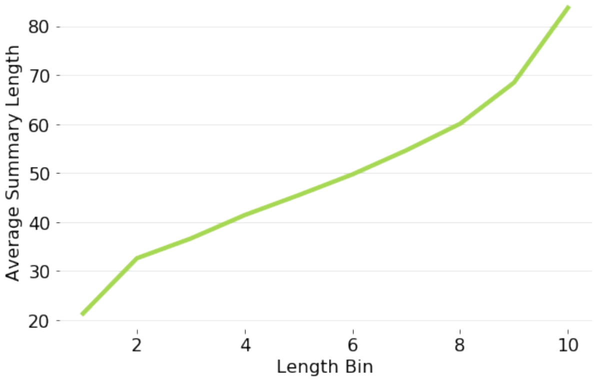

Figure 1: Effect of length control on summary length. The length variable can take 10 discrete values corresponding to length bins that divide the training set equally.

mentions the requested entity 57% of the time, while for all entities, the model mentions required entities only 30% of the time. The model might have difficulty generating summaries with entities which are unlikely to appear in the human references, e.g. entities far from the beginning of the article.

Source-style control is the least impactful control knob in terms of ROUGE, we report +0.2 F1-ROUGE1 in Table 3. Changing the source style variable indeed changes the summary as shown in Table 9. Generally, we observe that generated summaries in the Dailymail-style are more repetitive and slightly longer than the CNN-style summaries. This matches the characteristics distinguishing the two datasets in the reference text.

The impact of style requests might be greater with a richer set of styles. The dataset used for this experiments has only two available source-styles. In future work, we plan to evaluate on datasets where varied styles are readily available.

5.3 Summarization with Automatic Control

Our primary objective is to allow readers to control attributes of automatically generated summaries. However, we can also use our approach in the absence of reader desiderata by setting the control variables automatically. For length and source-style control, we set each variable to a constant value which maximizes ROUGE on the validation set. For entity control, we randomly sample an entity that appears in the first three sentences of the input document and provide it as the entity of interest.

Table 4 shows results on the entity-anonymized version of the dataset used by (Nallapati et al., 2016; Paulus et al., 2017) and Table 5 reports results on the original version of the dataset used by (See et al., 2017). In both cases, our method achieves a slight advantage over alternatives. On the original text, we report 39.75 F1-ROUGE1 as opposed to 39.53 for (See et al., 2017). On the entity-anonymized text, we report 38.68 F1-ROUGE1 as opposed to 38.30 for the best maximum likelihood training setting of (Paulus et al., 2017). Our model does not outperform the reinforcement learning model of (Paulus et al., 2017) which optimizes ROUGE. However, training objectives are orthogonal to our work on control variables and we expect their training objective to equally benefit our model. So far, we explored only maximum likelihood training and we defer ROUGE optimization to later work.

Overall, the improvement from automatic control is interesting as it shows that a better model might be obtained by providing extra information at training time. When the sequence-to-sequence model does not need to predict the summary length or the entity of interest, it can assign more capacity to generating text given these variables. This is particularly useful for variables which are hard to predict from the input due to intrinsic uncertainty, e.g. length. In later work, we are interested in exploring architectures which explicitly divide the prediction of control variables and sequence-to-sequence mapping.

6 Conclusion

In this work we presented a controllable summarization model that allows users to define high-level attributes of automatically generated summaries. We explore three variables: summary length, source-style, and entities of interest. We simulate user preferences for these variables by setting them to ground truth values which results in large ROUGE gains. The control variables are also effective without user input which we demonstrate by assigning them fixed values tuned on a held-out set. This slightly outperforms comparable state of the art summarization models. Target summary length is the most effective variable, followed by entities of interest and source-style. In future work we plan to provide control over additional attributes and explore the application of this technique to other text generation tasks.

Requested Length	Summary
Reference	Rabbit populations are rampant in greater @entity2 [Sydney] due to high summer rain. Public land managers are now desperate to reduce population numbers due to the millions of dollars worth of damage they cause. Carrots laced with calicivirus are likely to be scattered in public spaces for a second time this year after an initial attempt failed in some areas. About 70-100% of rabbits die once infected with calicivirus, which damages the animal's liver and gut and causes haemorrhaging and bleeding.
Length Bin 2	@entity0 [Easter] is over for the wild rabbits of greater @entity2 [Sydney] as councils and parks prepare another attempt to kill them off with a deadly virus. It comes after over 30 government bodies scattered carrots laced with calicivirus.
Length Bin 6	@entity0 [Easter] is over for the wild rabbits of greater @entity2 [Sydney] as councils and parks prepare another attempt to kill them off with a deadly virus. This year, because of really high summer rainfall - which led to great food availability - there has been a big surge in the rabbit population in @entity2 [Sydney].
Length Bin 10	@entity0 [Easter] is over for the wild rabbits of greater @entity2 [Sydney] as councils and parks prepare another attempt to kill them off with strategically placed carrots that have been laced with a deadly virus. This year, because of really high summer rainfall - which led to great food availability - there has been a big surge in the rabbit population in @entity2 [Sydney]. It comes after over 30 government bodies scattered carrots laced with calicivirus around public areas in March.

Table 7: Example summaries generated by changing the length control variable for the same input article.

Requested Entity	Summary
Reference	@entity94 [Theia], a bully breed mix, was apparently hit by a car, whacked with a hammer and buried in a field. "She's a true miracle dog and she deserves a good life," says @entity28 [Sara Mellado], who is looking for a home for @entity15 [Theia].
@entity26 [Moses Lake]	A stray pooch has used up at least three of her own after being hit by a car. She was taken in by **@entity26 [Moses Lake]**, @entity27 [Washington], resident @entity28 [Sara Mellado]. The dog's brush with death did not leave her unscathed, the dog managed to stagger to a nearby farm , dirt-covered and emaciated.
@entity28 [Sara Mellado]	A stray pooch in @entity5 [Washington State] has used up at least three of her own. She suffered a dislocated jaw , leg injuries and a caved-in sinus cavity – and still requires surgery to help her breathe. **@entity28 [Sara Mellado]** has set up a fundraising page to help meet the remaining cost of the dog 's care.
Reference	@entity1 [Linda MacDonald] was arrested Monday night after veering off the road and crashing her car into a wooden fence in @entity4 [Dummerston], @entity5 [Vermont]. The @entity15 [Shelburne], @entity16 [Vermont] woman claimed to have been talking on the phone and taking down directions when she crashed. Police smelled alcohol on her and when they administered a breathalyzer test, @entity20 [MacDonald] tested .02 per cent over the legal limit.
@entity17 [Route 5]	@entity1 [Linda MacDonald], 55 , was arrested for driving under the influence of alcohol Monday night in @entity4 [Dummerston], @entity5 [Vermont]. Police say the woman from @entity15 [Shelburne], @entity16 [Massachusetts] was driving drunk around 10:30pm when she ran off **@entity17 [Route 5]** in @entity4 [Dummerston].
@entity20 [MacDonald]	@entity1 [Linda MacDonald], 55 , was arrested for driving under the influence of alcohol Monday night in @entity4 [Dummerston], @entity5 [Vermont]. **@entity20 [MacDonald]** told officers that she crashed while talking on the phone and trying to take directions down on a legal note pad in her car . But when officers smelled alcohol on **@entity20 [MacDonald]**, they administered a breathalyzer test and she posted a .10 blood-alcohol content.

Table 8: Example summaries generated by specifying a different entity for the entity control variable. We present two examples for two different articles.

Requested Source Style	Summary
Reference (DailyMail)	@entity6 [Jared Forsyth], 33, joined @entity7 [Ocala Police Department] in 2012. He was shot in the chest by a colleague during firearms training Monday. @entity6 [Jared Forsyth] was rushed to hospital but died after undergoing surgery. Incident at @entity13 [Lowell Correctional Institution] under investigation. Officials say he was wearing a vest, but the round entered his arm.
CNN-Style	Officer @entity6 [Jared Forsyth], 33, had been a member of the @entity7 [Ocala Police Department] since 2012. He was wearing bulletproof vest, but round entered in his arm and went through his chest. @entity6 [Jared Forsyth] was rushed to hospital in critical condition.
Dailymail-Style	Officer @entity6 [Jared Forsyth], 33, had been a member of the @entity7 [Ocala Police Department] since 2012. He was rushed to @entity26 [Ocala Regional Medical Center] in critical condition and was taken into surgery. Police say the incident occurred about 3.30pm at a gun range at the @entity13 [Lowell Correctional Institution].
Reference (DailyMail)	@entity3 [Lois Remy] hit the decisive goal as @entity1 [Chelsea] beat @entity7 [Stoke] 2-1 on Saturday. He also scored the final goal in @entity1's [Chelsea's] 3-2 victory at @entity35 [Hull]. @entity0 [Jose Mourinho] praised @entity3's [Lois Remy's] scoring instinct and professionalism.
CNN-Style	[Chelsea] beat @entity7 [Stoke] 2-1 at @entity24 [Stamford Bridge] on Saturday. @entity3 [Lois Remy] scored his second consecutive goal in @entity1's [Chelsea's] win over @entity7 [Stoke]. @entity15 [Costa] hobbled off 10 minutes after replacing @entity16 [Oscar] at half-time.
DailyMail-Style	@entity1 [Chelsea] beat @entity7 [Stoke] 2-1 in their @entity12 [Premier League] clash on Saturday. @entity3 [Lois Remy] scored his second consecutive goal in the win against @entity7 [Stoke]. @entity15 [Costa] hobbled off 10 minutes after replacing @entity16 [Oscar] at half-time. @entity0 [Jose Mourinho] praised the professionalism and scoring ability of @entity3 [Lois Remy].

Table 9: Example summaries generated by changing the source style variable for the same input article.

Acknowledgments

We thank Yann Dauphin for helpful discussions. We thank Jonas Gehring and Denis Yarats for writing the fairseq toolkit used for these experiments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representation (ICLR)* .

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *CoNLL*.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)* .

R. Collobert, K. Kavukcuoglu, and C. Farabet. 2011. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*.

Dipanjan Das and André FT Martins. 2007. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU* 4:192–195.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks .

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. *Workshop on Stylistic Variations* abs/1707.02633.

Katja Filippova. 2017. Sentence and passage summarization for question answering. Talk given at EMNLP 2017 Workshop on New Frontiers in Summarization.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* .

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. 2017. Convolutional sequence to sequence learning .

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NIPS*.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Neural Information Processing Systems (NIPS)*.

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

Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. *Arxiv* abs/1312.6114.

Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Fader networks: Manipulating images by sliding attributes. *Arxiv* abs/1706.00409.

Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. 1990. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems (NIPS)*. pages 396–404.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out*.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development* 2(2).

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015a. Multi-task sequence to sequence learning. *Arxiv* abs/1511.06114.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. Effective approaches to attention-based neural machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kathleen McKeown. 1992. *Text generation*. Cambridge University Press.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *Conference of the Association for the Advancement of Artificial Intelligence (AAAI)* .

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *Conference of the Association for the Advancement of Artificial Intelligence (CONLL)* .

Ani Nenkova, Kathleen McKeown, et al. 2011. Automatic summarization. *Foundations and Trends® in Information Retrieval* 5(2–3):103–233.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *ICML*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304* .

Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Joseph Pal, and Aaron C. Courville. 2017. Adversarial generation of natural language. *Workshop on Representation Learning for NLP* .

Alexander M Rush, SEAS Harvard, Sumit Chopra, and Jason Weston. 2015. A neural attention model for sentence summarization. *Conference on Empirical Methods in Natural Language Processing (EMNLP)* .

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *Annual Meeting of the Association for Computational Linguistics (ACL)* .

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. *Annual Meeting of the Association for Computational Linguistics (ACL)* .

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. *Arxiv* abs/1705.09655.

Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. 2013. On the importance of initialization and momentum in deep learning. In *ICML*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Neural Information Processing Systems (NIPS)*.

Jun Suzuki and Masaaki Nagata. 2017. Cutting-off redundant repeating generations for neural abstractive summarization. In *European Conference of the Association of Computer Linguists (EACL)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* .

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015a. Pointer networks. In *Conference on Advances in Neural Information Processing Systems (NIPS)*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015b. Show and tell: A neural image caption generator. *Conference on Computer Vision and Pattern Recognition (CVPR)* .

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*.

Junbo Zhao, Y. Kim, K. Zhang, A. M. Rush, and Y. LeCun. 2017. Adversarially Regularized Autoencoders for Generating Discrete Structures. *ArXiv* 1706.04223.