Sequentially Controlled Text Generation

Anonymous ACL submission

Abstract

While GPT2 generates sentences that are remarkably human-like, longer documents can ramble and do not follow human-like writing structure. We study the problem of imposing structure on long-range text. We propose a novel controlled text generation task, *sequentially controlled text generation*, and identify a dataset, *NewsDiscourse* as a starting point for this task. We develop a sequential controlled text generation pipeline with generation and editing. We test different degrees of structural awareness and show that, in general, more structural awareness results in higher control-accuracy, grammaticality, coherency and topicality, approaching human-level writing performance.

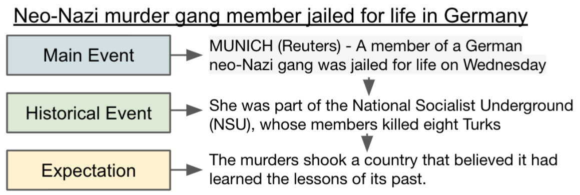


Figure 1: We study the task of *sequentially-controlled generation*: generating documents exhibiting structure given by a sequence of local control codes. Shown is a news article with it's Van Dijk structure (Van Dijk, 2013) and headline. Our models take as input the headline and discourse tags and generate a sequence of sentences. We explore the degree of structural awareness (local, past-aware or full-sequence) for controlling each sentence in the document, with the goal of generating the most structurally faithful, coherent and topical text.

1 Introduction

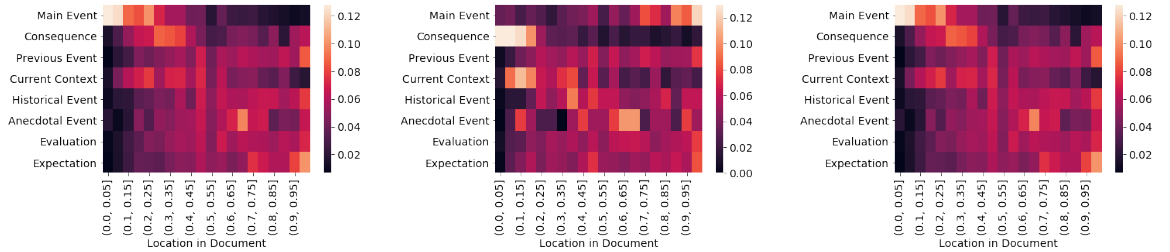
Imagine that you are tasked with: Write a "Related Works" section. Would it help to know the *past structure* of the article (e.g. it is coming after the "Discussion" section)? How about the *full structure* of the article (e.g. after the "Introduction" but before the "Problem Statement")?

The macro-structure of text (i.e. it's discourse structure (Po¨ ttker, 2003)) impacts both human and machine comprehension (Emde et al., 2016; Sternadori and Wise, 2010; Lu et al., 2019; Zhou et al., 2020). Although naive language models have made impressive advancements and generate fluent text (Radford et al., 2019; Brown et al., 2020; Beltagy et al., 2020), the text is *structurally* dissimilar to human-written text (Figure 2, Section 7). Even the well-known Ovid's Unicorn generation, which seems like a natural news article, exhibits unnatural structure (see Appendix F).

On the other hand, although numerous works have focused on content planning using keywords (Yao et al., 2019), plot-design (Rashkin et al., 2020) and entity tracking (Peng et al., 2021), macro-structural control has been relatively understudied.

So, we study *(1) how to impose macro-structural control on narrative text generation and (2) how much structural awareness during generation contributes to well-structured and fluent text.* We propose a novel task, *sequentially controlled text generation*. In this task, the user provides a sequence of local control codes, each guiding the generation of a sentence. In our experiments, we use headlines as prompts and Van Dijk (2013) discourse tags as control codes (Figure 1).

We develop methods to solve this task, expanding prior work focused on *single control code generation* (Keskar et al., 2019; Dathathri et al., 2019; Yang and Klein, 2021). Because our methods allow us to probe the dependencies between tag sequences, which prior methods did not, we are able to test what degree of structural awareness yields the highest-quality documents: **local-only** (where the generator is only aware of the current sentences' control code), **past-aware** (where the generator is aware of the current sentences' control code *and* all previous control codes), and **full-sequence** (where the generator is aware of the entire document's sequence of control codes). We show that more struc-



(a) Structure of human-written articles. (b) Structure of naively generated GPT2 articles (c) Structure of sequentially controlled GPT2 articles.

Figure 2: Discourse structure (Van Dijk, 2013) of articles generated according to different processes. The likelihood of a tag in the kth fraction of a news article is shown. Machine-generated structure is labeled by humans.

tural awareness, especially of past structure, helps us generate the highest-quality text. Finally, we show how to further balance *structural* and *local* control in a pipeline by combining the structurally-aware generation methods described above with a local sentence-level editing technique. Using both techniques in tandem generates fluent documents that exhibit appropriate structure.

In summary, our novel contributions are:

- We propose a novel task, *sequentially controlled text generation* and identify a discourse schema (Van Dijk, 2013) and dataset (Choubey et al., 2020) to explore this task (Section 2, 4).
- We combine two different approaches in controlled text generation: *generation* and *editing*, and show that the highest-quality text is generated when both of these approaches are used (Section 3).
- We use our methods to study the *degree* of structural control yields the highest-quality text: *local*, *past-aware* and *full-sequence* control. We show that overall, *full-sequence* produces optimal text over an array of metrics (Section 7).

We hope in the future that this work will provide a natural complement to other forms of controlled generation, like fact-aware generation (Logan IV et al., 2019). We envision this line of work being used by journalists to quickly prototype different structures for their work, or fill in missing structural components to aid in human-in-the-loop computational journalism (Cohen et al., 2011).

2 Problem Statement

We assume, as input, a headline sentence, X_0, and a sequence of control codes $\mathbf{c} = c_1, ..., c_S$ of length S (i.e. one for each sentence we wish to generate in the document. *Adjacent codes can be of the same type*.) We wish to produce, as output, a document X of length S as a sequence of sentences $X = X_1, ..., X_S$, each composed of a sequence of words $X_k = x_1, ..., x_{n_k}$ of length n_k.

We define the sequentially controlled text generation objective as:

$$p(x|\mathbf{c}) = \prod_{k=1}^{S} \prod_{i=1}^{n_k} \underbrace{p(x_i|x_{<i}, X_{<k}, \mathbf{c})}_{t_1\text{: word likelihood}} \quad (1)$$

Where x_i is a word in sentence k, $x_{<i}$ are the preceding words, $X_{<k}$ are the preceding sentences (including the headline, X_0). c_k is the control code for k. We assume that $\mathbf{c}$, the entire sequence of control-codes for a document, is given.

We use Bayes rule to factorize t_1 into:

$$\propto \underbrace{p(x_i|x_{<i}, X_{<k})}_{t_2\text{: naive word likelihood}} \underbrace{p(\mathbf{c}|x_i, x_{\leq i}, X_{<k})}_{t_3\text{: class likelihood}} \quad (2)$$

t_2 is calculated using a standard pretrained language model (PTLM) and t_3 is calculated by a trained discriminator. This allows us to maximally re-use naively trained language models and, we show, is far more resource efficient than fine-tuning a prompt-based model.

Three approximations for t_3 are:

$$\textbf{Local-Only} \qquad t_3 \approx p(c_s|x_i, x_{<i}, X_{<s}) \quad (3)$$

In the local-only model, we assume each control code c_k is conditionally independent of other control codes given x_i. Thus, our generator model t_1 is made aware only of local structure: the control code c_k pertaining to the current sentence, k. Because of this conditional independence assumption, *local-only* control is similar to prior work that used only single-control codes, where the goal was to generate a single sentence $p(x|c) = \prod_{i=1}^{n} p(x_i|c)$ (Keskar et al., 2019). However, we show that we can remove these independence assumptions and study more complicated structural control which, we show later, produces more coherent output.

Past-Aware

$$t_3 \quad \prod_{j=1} p(c_j|x_i, x_{<i}, X_{<k}, c_{<j}) \quad (4)$$

In the past-aware model, we assume autoregressive dependence between control codes, conditioned on x. Control codes for future sentences, $c_{>k}$, are conditionally independent. In Equation 1, this results in x_i being dependent on c_k and the sequence of control codes, $c_{<k}$.

Full-Sequence

$$t_3 = \prod_{j=1} p(c_j|x_i, x_{<i}, X_{<k}, c_{<j}) \quad (5)$$

In the full-sequence model, we make no conditional independence assumptions.

We can restrict both the past-aware and the full-sequence approximations to a sliding window around sentence s[1]. We can also add a prior on $p(\mathbf{c})$ to induce a discount factor[2]. This focuses the generator on control code c_k and down-weights surrounding control codes.

In the next section, we show how to model these objectives. We first describe the discriminator we use as our control-code model, the controlled generation techniques and the editing techniques we adapt.

3 Methodology

As described in Section 2, we can efficiently do generation by combining a *naively-trained* language model with a discriminator. *Hence, the discriminator is the main architectural component that allows us to incorporate inter-dependencies between control code sequences.* We start by describing how our discriminator models different degrees of structural awareness (Equations 3, 4 and 5) in Section 3.1.

We design a generation pipeline to balance *structural* and *local awareness*. The flow we use to accomplish this is depicted in Figure 3. The first step is **Generation**. Here, we sample each word, x_i using techniques described in Section 3.2 which allow us to leverage our discriminator to impose *structural control*. When we have completed a sentence, we move to **Editing**. Here, we edit the sentence to further impose *local control* on each sentence, updating x to optimize a variation of Equation 1: $p(x_i|x_{-i}, c_k)$, discussed in Section 3.3.

3.1 Discriminator

The discriminator we construct takes as input a sequence of sentences ($\mathbf{X}$) and a sequence of local control tags ($\mathbf{c}$). Our architecture combines a sentence-classification model, similar to that used in Spangher et al. (2021), with separate a label embedding architecture to incorporate knowledge of $c_{<j}$. Hence, we can make predictions for c_j based not only on x, but prior tags, $c_{<j}$, allowing us to model structural dependencies (Equation 2). For a full description of architecture, see Appendix A.

We train it to model local-only, past-aware and full-sequence control variants expressed in Section 2 (Equations 3, 4 and 5): we train separate prediction heads to make predictions on $c_{k-w}, ..., c_k, ..., c_{k+w}$, i.e. labels from $-w, ..., +w$ steps away from current sentence k. For local-only control (Equation 3) we only use predicted probabilities from the main head, k. In past-aware control (Equation 4), we multiply predicted probabilities from heads prior to the current sentence $< k$, and for **full-sequence** control, we multiply predicted probabilities from all heads.[3] We now describe how we use these predictions.

3.2 Generation

We combine our discriminator's predictions with a naive PTLM to solve Equation 2 in two different ways: **Hidden-State Control**, based on Dathathri et al. (2019) and **Direct Probability**, based on Yang and Klein (2021).

Hidden-State Control (HSC): Wolf et al. (2019)'s GPT2 implementation caches hidden states H to produce logits approximating $p(x_i|x_{<i})$. We perturb these hidden states H, re-

[1] i.e. t_3 ranges only from $j = k - w ... k + w$ instead of the full sequence of sentences. In practice, we use $w = 3$.

[2] The form of our prior is: $t_3 = \prod_{j=1}^{S} m(i,j) p(c_j|x_i, x_{<i}, X_{<k}, c_{<j})$, where $m(i,j) = b^{|i-j|}$. We experiment with $b = [.33, .66, 1]$.

[3] For the editing operation, the discriminator is trained without the contexualizing layer (i.e. Transformer and a_i layers are not used) because gradients need to be computed that pertain only to the sentence being edited, not previous sentences.

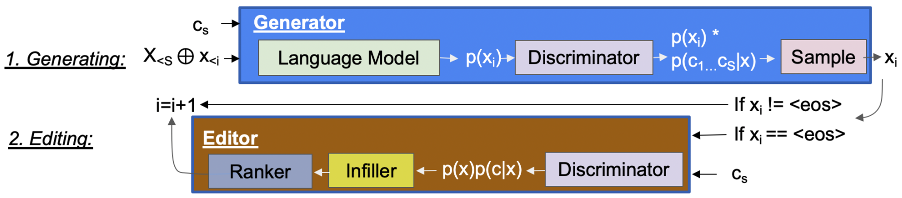


Figure 3: **Generation process**. First, we perturb the output of a language model using a structurally-aware classifier to approximate $p(x_i|x_{<i}, X_{<k})p(\hat{c}|x_{<i}, X_{<k})$ and generate word x_i by sampling from the perturbed distribution . When we generate an $<eos>$ token, we edit the sentence. We use a discriminator to identify class-salient words to mask, generating masked sentence M, and infill to boost class likelihood.

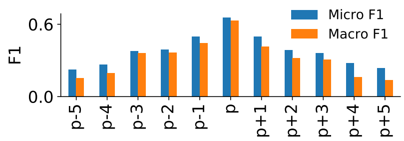


Figure 4: **Discriminator performance** on test data. F1 scores for $p(c_j|X_{<k}, x_{<i}, c_{<j})$ predictions. Sentence index k and word index i are fixed: we show error for using the current sentence to predict all past, current and future labels.

sulting in $\hat{H}$ that produce logits approximating Equation 1 instead. We generate H from a naive PTLM and using this to make a prediction $\hat{c}$ using our discriminator. We then calculate the loss $L(\hat{c}, c)$ and backpropogate to H to derive $\hat{H}$.

Direct-Probability Control (DPC): We calculate $p(x_i|x_{<i}, X_{<s})$ to identify the 200 most likely x_i under the naive language model, $\{x_{i,j}\}_{j=0}^{200}$. Then we calculate $p(c_s|x_{i,j}, x_{<i}, X_{<s}, c_{<s})$ for each $x_{i,j}$ using our discriminator. We directly multiply these probabilities to calculate Equation 1[4].

Note that the HSC and DPC algorithms are extensions of previous work: the difference is that here they are used to model control code *sequences* rather than *single* tags. *The key components that allow this is our discriminator, which makes predictions based on label sequences, and our algorithm which, as shown in Figure 3, increments codes each time an <eos> token is generated.*

3.3 Editing

After we have finished generating a sentence, we edit it to introduce more discourse markers of the local control code.

We identify words in our input sequence that have the most impact on control-code prediction by using the gradient on our input sentence of the discriminator's loss, following Ross et al. (2021). We use only the current sentence prediction made by our discriminator (i.e. Equation 3), so that we impose local control on the sequence even in settings where the generator imposes structural control.

We cull the high-gradient words based on heuristics[5] to encourage the editor to introduce explicit discourse markers. We fine-tune a label-aware infilling model (Raffel et al., 2019) to generate candidate edits[6] given the masked input. We mask and infill until we have generated a sentence that has an increased likelihood $p(c_k|\hat{x}_k) > p(c_k|x_k)$, and generate edit candidates ($n = 10$). We select edits on the basis class likelihood and perplexity[7].

For more comparison and distinction from previous work for both Generation and Editing, see Appendix D.1, E.

4 Datasets and Schema

The form of sequential control we study is *discourse*: i.e. the functional role sentences play in a document's larger argumentative purpose. We use a news discourse schema proposed by Van Dijk (2013). In Choubey et al. (2020), authors apply this schema and annotate a dataset, *NewsDiscourse*, consisting of 802 articles from 3 outlets[8], tagged on the sentence level. Their schema consists of 9 classes: { Main Event, Consequence, Current Con-

[4]Note that DPC has the advantage of being simpler to implement and batch-parallelizable. However, the restriction to the top $k = 200$ words selected according to $p(x_i|x_{<i}, X_{<s})$ means that we might be limiting discriminator perturbation of word-selection.

[5]Words that are *not* proper nouns, named entities (except the DATE class) or adjectives, as we find these categories are more likely to be topic words spuriously correlated with control-codes.

[6]A T5 model trained using a specific input template incorporating the label. E.g. `label: Background. text: The senator <MASK> to the courtroom to <MASK>.`

[7]Perplexity of the entire generated document so far is used as a selection criteria, $PPL(x_k \oplus X_{<k})$, to encourage edits preserving the logical flow of the document.

[8]nytimes.com, reuters.com and xinhuanet.com

Headline	UPDATE 1-Belarus food retailer lines up landmark overseas IPO
Main Event	SILVER SPRING, Colo. (AP) — A company in Finland is poised to list its first international grocery chain in the United States after a yearlong search, raising concerns about potential corporate conflicts of interest and legal issues around using foreign intellectual property.<lendoftextl>
Previous Event	The world's largest retailer of meat and fish products hopes to become one day the leader in convenience groceries.<lendoftextl>
Expectation	It says it could become a leader in fresh fruits and vegetables, as well.<lendoftextl>
Evaluation	About 300 Finnish companies have applied to list on the New York Stock Exchange under a new plan to bypass the U. S. government and create "licenses" that allow them in many other countries, according to a regulatory filing.<lendoftextl>
Evaluation	The European Union prohibits using foreign intellectual property for purposes outside the country, and Finland's National Stock Exchange does not require approval before selling a company'lls share in its new company.<lendoftextl>
Evaluation	"What's at stake here is transparency, a sense of fairness to all the stakeholders.<lendoftextl>
Evaluation	I don't think it's right for companies to have intellectual property rights," says Michael Vakilainen, an independent analyst who has tracked the company since 2008.<lendoftextl>
Expectation	He says there are potential conflicts of interest, because one partner is the government.<lendoftextl>
Expectation	"What if you're a government contractor?"<lendoftextl>

Table 1: Sample document generated. Generation Method = Direct Prob. Control. Structure = Past Aware. Edited = False. (Hyperparams = $\quad = :75$, $b = :33$)

text, Previous Event, Historical Event, Anecdotal Event, Evaluation, Expectation g.[9]. *Although each sentence is tagged with a code, codes often repeat.* For example, an entire paragraph can be tagged with Main Event sentences. We show a partial sample in Figure 1. We adopt this schema to describe each news article's structure.

We also use a dataset of unlabeled news articles[10] to fine-tune GPT2 model for news. We sample 30,000 documents from this dataset in a manner so that the distribution of sentence-lengths matches the distribution of sentence lengths in the Choubey et al. (2020) dataset.

5 Implementation Details

We fine-tune a `GPT2-base` model on a large news corpora with a max word-piece length=2048[11]. We use this to generate naive PTLM language-modeling *as well as* sentence-embeddings in our Discrimination model. Further implementation details discussed in Appendix A.

We discuss the discriminator results here briefly. As shown in Figure 4, the primary head, p, has a Micro F1-score of $:65$, which approaches state-of-the-art on this dataset[12]. However, performance degrades rapidly for heads farther from p. For more results on discriminator performance, including experimental variations, see Appendix A.1.

[9] For a detailed class description, see Appendix F.1

[10] `kaggle.com/snapcrack/all-the-news`. Dataset originally collected from `archive.org`. We filter to articles from nytimes.com and reuters.com.

[11] Rather than 1024 in (Radford et al., 2019). We observe that $>99\%$ of human-generated news articles were shorter than 2048 word pieces.

[12] $:71$ Micro-F1 in Spangher et al. (2021), which used auxiliary datasets.

6 Experiments

We sample 10 documents from the test set of our discourse dataset ($n = 200$) to test different pipeline settings. The input to our models is **a headline (as a prompt) and the *full sequence of gold-truth discourse labels*** of that document.

Baselines We compare our experimental pipelines (Section 3) with the following baselines: (1) **Naive GPT2** generation given only the headline as input (i.e. no control codes), (2) a fine-tuned **Prompting** approach and (3) the original **Human**-written articles.

For (2), we directly train a class-conditional language model to generate text by including labels in the prompt, as in Keskar et al. (2019). Local-only prompting is achieved by only including the local control code (and prior generated sentences) in the prompt, and updating the prompt to generate a new sentence. For past-aware prompting, we include all control codes prior to our current sentence in the prompt, and update on every new sentence. Finally, for full-sequence prompting, we including the full sequence of control codes in the prompt. (See Appendix C for more details and examples of prompt design.)

For each of these baselines, we test with and without editing (with the human-written text being edited by our algorithm in **Human** and with the generated text in all other trials being edited).

Evaluation For all pipelines, we select the best hyperparameter con gurations based on perplexity and model-assigned class likelihood. Then, we manually annotate each generated document for 4 metrics: Accuracy (0-1)[13], Grammar (1-5)[14], Logical Flow (1-5)[15] and Topicality (1-5)[16]. We recruit two expert annotators with journalism experience to perform annotations blindly without awareness to which generation pipeline was used, and nd moderate agreement 2 [:36; :55] across all categories. For more details, see Appendix G. We record model-dependent and non-model automatic metrics used by See et al. (2019), described further in Appendix B.

(a) (b)

(c) (d)

Figure 5: Comparison of different structural control methods across different pipelines and hyperparameters.

7 Results

Best Overall Trial We show automatic and human metrics for the subset of pipelines with top-performing hyperparameters in Table 2. In general, the highest-performing generation pipelines are all variations of DPC with either past-aware, or full-sequence structural control.

We observe that DPC with past-aware control and editing has the highest class-label accuracy, nearly approaching the human trials. The top performing pipeline for logical ow is also DPC with past-aware control, but without editing. And the top performing pipelines for grammar and topicality are DPC with full-Sequence control and without editing.

(a) (b)

(c) (d)

Figure 6: The effect of editing, across different pipelines and hyperparameters.

Effect of Different Pipeline Components We show the distributional shifts in performance across all trials, in Figures 5, 6[17]. Structural control has a largely positive effect on generated text. In Figure 5, we nd that Full-Sequence models are, on average, able to generate the most label-accurate sentences with the best grammar, logical ow and topicality. Finally, editing improves accuracy, grammar and logical ow (Figure 6.)

The original human-generated text is our gold-standard, and it is highly class-accurate, grammatical, coherent and topical. Interestingly, as seen in Table 2, editing can also be applied to human-written text to boost label accuracy, but at the expense of coherence.

8 Discussion

We set out to answer two questions in this research: (1) whether we could impose structural control over generated documents and (2) what kinds of structural control (local-only, past-aware, or full-sequence) had the greatest effect on discourse, ow, topicality and grammaticality. Our novel pipelines, which extend various discriminator-based approaches for generation and editing, approach human-level performance. However, a gap still remains, suggesting the need for more research or data collection.

Insight #1: Some structural information improves all metrics of quality. Our structural exploration suggests that, for the best-performing pipelines, *past* structural information (along with editing) boosts class accuracy the most, but knowledge of the full-sequence does not. In the analogy given in the Introduction, this equates to: to write a "Related Works" section, it helps to know that

[13] Accuracy: how close a generated sentence matches the discourse function of the gold-truth label for that sentence.

[14] Grammar: how grammatical and locally coherent a sentence is

[15] Logical Flow: how well a sentence functions in the ow of the story

[16] How well each sentence corresponds to the original headline of the article.

[17] And 10, in Appendix E.

Gener-ation	Struct-ure	Label Acc. ↑ (0-100)	Gram-mar ↑ (1-5)	Logical Flow ↑ (1-5)	On-Topic ↑ (1-5)	Perplex.↓	Diverse Ngrams ↑ (%)	Sent. Len.**	Unseen Words ↓ (%)
		Human-Annotated Metrics				Automatic Metrics			
Naive GPT2		20.0/64.4	4.2/4.5	4.7/4.3	4.6/4.2	48.2/45.4	7.1/8.3	24.9/38.8	4.7/3.2
Gen-Base: Prompt	local	22.2/51.1	2.8/3.9	2.4/3.0	2.3/2.8	24.4/43.4	3.7/6.5	39.7/32.4	10.6/8.7
	past	20.0/31.1	2.9/3.6	2.4/2.9	2.3/3.7	52.2/32.0	5.0/4.5	35.0/44.5	9.3/7.1
	full	46.7/64.4	4.4/4.4	3.6/3.7	3.9/3.5	42.5/49.2	7.3/7.8	35.5/42.6	4.6/4.9
Method #1: HSC	local	28.9/42.2	3.3/3.7	2.7/3.2	3.1/3.4	246.4/115.5	7.0/6.9	16.2/17.5	8.0/6.9
	past	44.4/60.0	3.4/3.8	3.0/3.0	3.2/3.3	178.3/147.4	7.5/7.5	14.8/18.8	8.1/6.7
	full	55.6/68.9	3.5/4.2	4.0/3.7	4.2/4.3	134.5/129.6	7.2/7.8	17.3/20.7	7.0/7.1
Method #2: DPC	local	44.4/64.4	4.0/4.4	3.6/4.1	3.8/3.5	42.1/39.9	5.8/8.3	24.8/42.6	4.7/3.0
	past	64.4/68.9	4.5/4.6	4.4/4.3	4.4/4.5	37.0/42.2	7.9/8.4	33.1/42.7	3.9/3.1
	full	66.7/68.9	4.7/4.5	4.3/4.3	4.7/4.4	42.3/45.6	8.0/8.1	28.2/40.4	4.3/3.3
Human		93.3/95.6	4.9/4.7	4.9/4.7	4.9/4.9	34.2/41.0	8.7/8.7	37.9/39.6	4.2/4.5

Table 2: Metrics on different trial runs. Each cell shows Unedited/Edited variants. (Hyperparams = .75, b = .33). ** Optimal sentence length is determined relative human generation, i.e. min|x − 37.9|.

it comes after the "Introduction" vs. the "Discussion", but not information of what sections come after. This is perhaps because enough signal is already given by the past sequence and the full sequence just adds more noise. However, full sequence information does yield the best grammar and topicality. This might indicate a regularizing role played by the full-sequence. In general, we suspect that past-aware modeling and editing both push the model more towards the class label at the expense of topicality, ow and grammar, while full-sequence does the opposite. In practice, some combination of these pipeline components might be desired.

Insight #2: Weak discriminators can still impose accurate control. At .61 macro F1, our discriminator is a relatively weak classi er. Previous work in classi er-based controlled text generation used large training datasets and classi ers that routinely scored above .8 F1 (Dathathri et al., 2019; Yang and Klein, 2021). The weakness of our discriminator is one reason why HSC may have performed poorly. However, in other trials we see strong accuracy. Thus, even with a weak classi er, we can control generation. This might be because even a weak discriminator can still give relative differences between generation that does or does match the control code.

Insight #3: Evaluating text candidates using multiple model's perplexity might result in better selections. Just as surprisingly, editing also has an overall average positive effect on generation accuracy and generation quality (Figure 6). We had hypothesized that, because editor makes locally-aware in lling decisions, it would improve class-accuracy but hurt other metrics of document quality, like topicality and ow. Indeed, for the top-performing trials, like DPC and Human, Editing only improves class accuracy. However, grammar and ow improves in other trials. This could be because, as mentioned in Section 3.3, we selected candidates based on how well they makes sense in the document. This also suggests that using multiple PTLMs to select for better quality combines different virtues of each model.

Error Analysis: We observed that sentence tokenizing remained a huge challenge. Many of the grammar errors that our annotators observed were from sentences that ended early, i.e. after decimal points. Indeed, the correlation between sentence-length and grammar is relatively high (r = .34). One reason for this could be that error-prone sentence tokenizing models provided faulty training data during pretrainining of LMs. This will continue to hinder document-level structural work, which often relies on a model accurately ending a sentence. Another observation, in Table 2, is that perplexity doesn't necessarily correlate with human judgements of quality, especially for more complex writing like Financial news reporting.

9 Related Work

Discourse-Aware Narrative Text Generation. Generating narrative text, such as news articles and scientific reports, has been a long standing problem in NLP. Early work relies on template (Xu et al., 2018; Wiseman et al., 2018), rules (Ahn et al., 2016; Leppänen and Toivonen, 2021), or specialized architecture (Fan et al., 2018; Bosselut et al., 2018) that are hard to generalize. Recently, pre-trained Transformers have shown impressive capabilities to produce fluent text, yet it is unclear how to adapt them to document-level generation with appropriate discourse structures.

Controlled Generation The black-box nature of neural generation models posts challenges for many real-world applications (Wiseman et al., 2017; Holtzman et al., 2019). Researchers have designed various techniques to control the syntactic structure (Goyal and Durrett, 2020), sentiment (Hu et al., 2017; Luo et al., 2019), and language style (Niu and Bansal, 2018; Cao and Wang, 2021). Most notably, the CTRL model (Keskar et al., 2019) conditions the output by incorporating textual control codes during the pre-training stage. However, such training is resource-intensive and requires large datasets. Alternatively, PPLM (Dathathri et al., 2019) and FUDGE (Yang and Klein, 2021) achieve inference-time control through either directly manipulating the generator's hidden states, or adjusting the probabilistic distribution over the output vocabulary. Our work differs from prior work in that we tackle structured control instead of a single attribute.

Sequentially Controlled Generation Sequential control for text generation has been explored from many angles, from symbolic planning approaches (Meehan, 1976; Lebowitz, 1987), to keyword-based approaches (Yao et al., 2019) and concept, event and entity driven planning approaches (Rashkin et al., 2020; Peng et al., 2021; Alabdulkarim et al., 2021). We are the first, to our knowledge, to utilize a purely latent control structure based off of discourse structures. There is increasing interest in exploring how discourse can be used to guide generation (Ghazvininejad et al., 2021; Cohan et al., 2018), from early works developing discourse schemas for generation (Mann, 1984; Stede and Umbach, 1998) to evaluating creative generation pipelines (Hua and Wang, 2020). However, neither direction allows discourse structures to be explicitly controlled in generation.

Editing. Most existing neural models generate text in one-shot, from left to right. Recently, an emerging line of research (Guu et al., 2018; Malmi et al., 2019; Kasner and Dušek, 2020) has explored editing as part of the generation pipeline to further improve the output quality, or satisfy certain desired constraints. Our work builds off of the MiCE framework (Ross et al., 2021), which was originally designed for generating contrastive explanations.

Finally, we see overlaps as well to an earlier paradigm of generative modeling: Bayesian models for text like Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and, more interestingly, sequential variants (Du et al., 2012). There is recent work marrying PPLM-style controlled text generation with topic modeling (Carbone and Sarti, 2020). Such directions might lead to more hierarchical, structural control.

10 Conclusion

We have formalized a novel direction in controlled text generation: sequentially controlled text generation. We extended different techniques in controlled text generation to fit this direction, and have shown how a news discourse dataset can be used to produce news articles exhibiting human-like structure. We have explored what degrees of structural awareness yield the most human-like output: more structural control yields higher-quality output. And, we shown how to combine structural control with local editing. We have probed different parts of our pipeline to show the effects of each part.

11 Ethics Statement

11.1 Limitations

A central limitation to our work is that the datasets we used to train our models are all in English. As mentioned previously, we used Choubey et al. (2020)'s NewsDiscourse dataset, which consists of the sources: nytimes.com, reuters.com and xinhuanet.com. Although xinhuanet.com is a Chinese source, they used English-language articles. Additionally, we used an unlabeled news dataset from Kaggle[18] for fine-tuning GPT2-base and for calculating some automatic metrics like Unseen Words We filtered this dataset down to two English-languge, Western domains: nytimes.com and reuters.com in order to match the domains are closely as possible to the NewsDiscourse dataset.

Thus, we must view our work in discourse generation with the important caveat that non-Western news outlets may not follow the same discourse

[18] kaggle.com/snapcrack/all-the-news

structures in writing their news articles. We are not aware of existing Van Dijk-style (Van Dijk, 2013) datasets towards which we could provide an exact comparison. But, we hope in future work to look at other kinds of discourse structures that might exist in other languages.

11.2 Risks

There is a risk that the work will be used for misinformation or disinformation. This risk is acute in the news domain, where fake news outlets peddle false stories that attempt to look true (Boyd et al.; Spangher et al., 2020). Along this vein, there is the aforementioned work using discourse-structure to identify misinformation (Abbas, 2020; Zhou et al., 2020), and the risk in developing better discourse-aware generation tools is that these misinformation detectors might lose their effectiveness.

There is also a non-malicious misinformation risk, as large language models have been known to generate hallucinated information (Choubey et al., 2021). The more such threads of research are pursued without an accompanying focus on factuality and truth, the more risk we run of polluting the information ecosystem. However, like others (Dathathri et al., 2019), we see a value in continuing this direction of research, even if this current work is not the nal output we wish to see being used by non-researchers in the world. It is one step along the way.

There is also a risk that news articles in either of our datasets contain potentially libelious or defamatory information that had been removed from the publishers' website after the dataset was collected. However, we do not release either of the datasets we use, so we do not see our actions as privacy-violating.

11.3 Licensing

Of the two datasets we used, NewsDiscourse (Choubey et al., 2020) is published as a dataset resource in ACL 2020. They collected reuters.com and xinhua.net via crawling, and the nytimes.com from existing academically licensed datasets (Bhatia et al., 2015; Sandhaus, 2008).

We were unable to ascertain the license for the Kaggle dataset. It has been widely used in the academic literature, including in papers published in ACL venues (Pathak and Srihari, 2019) and others (Alhuqail, 2021). We corresponded with the authors and opened a discussion question [URL withheld to preserve anonymity] seeking more information about the license. The authors are public about their desire to have their dataset used[19] and we have had independent lawyers at a major media company ascertain that this dataset was low risk for copyright infringement.

11.4 Computational Resources

The experiments in our paper required computational resources. We used 8 30GB NVIDIA GPUs, AWS storage and CPU capabilities. We designed all our models to run on 1 GPU, so they did not need to utilize model or data-parallelism. However, we still need to recognize that not all researchers have access to this type of equipment. We used Huggingface GPT2-base models for our predictive tasks, and will release the code of all the custom architectures that we constructed. Our models do not exceed 300 million parameters.

11.5 Annotators

We recruited annotators from professional networks. Both consented to annotate as part of the experiment in exchange for acknowledgement. One is a graduate student studying in Europe, and the other is a former journalist. One annotator is female, and the other is male. One is half-Asian and half-white identifying, the other is white. Both identify as cis-gender. This work passed IRB.

References

Ali Haif Abbas. 2020. Politicizing the Pandemic: A Schemata Analysis of COVID-19 News in Two Selected Newspapers. *International Journal for the Semiotics of Law-Revue internationale de Sémiotique juridique*, pages 1–20.

Emily Ahn, Fabrizio Morbini, and Andrew Gordon. 2016. Improving uency in narrative text generation with grammatical transformations and probabilistic parsing. In *Proceedings of the 9th International Natural Language Generation conference*, pages 70–73, Edinburgh, UK. Association for Computational Linguistics.

Amal Alabdulkarim, Winston Li, Lara J Martin, and Mark O Riedl. 2021. Goal-directed story generation: Augmenting generative language models with reinforcement learning. *arXiv preprint arXiv:2112.08593*.

Noura Khalid Alhuqail. 2021. Author identi cation based on nlp. *European Journal of Computer Science and Information Technology*, 9(1):1–26.

[19] https://components.one/datasets/all-the-news-2-news-articles-dataset/

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from rst discourse parsing. arXiv preprint arXiv:1509.01599.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. the Journal of machine Learning research, 3:993–1022.

Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po-Sen Huang, and Yejin Choi. 2018. Discourse-aware neural rewards for coherent text generation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 173–184, New Orleans, Louisiana. Association for Computational Linguistics.

Ryan L Boyd, Alexander Spangher, Adam Fourney, Besmira Nushi, Gireeja Ranade, James Pennebaker, and Eric Horvitz. Characterizing the internet research agency's social media operations during the 2016 us presidential election using linguistic analyses.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Shuyang Cao and Lu Wang. 2021. Inference time style control for summarization. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5942–5953, Online. Association for Computational Linguistics.

Ginevra Carbone and Gabriele Sarti. 2020. Etc-nlg: End-to-end topic-conditioned natural language generation. arXiv preprint arXiv:2008.10875.

Prafulla Kumar Choubey, Aaron Lee, Ruihong Huang, and Lu Wang. 2020. Discourse as a Function of Event: Profiling Discourse Structure in News Articles around the Main Event. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5374–5386, Online. Association for Computational Linguistics.

Prafulla Kumar Choubey, Jesse Vig, Wenhao Liu, and Nazneen Fatema Rajani. 2021. Mofe: Mixture of factual experts for controlling hallucinations in abstractive summarization. arXiv preprint arXiv:2110.07166.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Sarah Cohen, James T Hamilton, and Fred Turner. 2011. Computational journalism. Communications of the ACM, 54(10):66–71.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. arXiv preprint arXiv:1912.02164.

Lan Du, Wray Buntine, Huidong Jin, and Changyou Chen. 2012. Sequential latent dirichlet allocation. Knowledge and information systems, 31(3):475–503.

Katharina Emde, Christoph Klimmt, and Daniela M Schluetz. 2016. Does storytelling help adolescents to process the news? a comparison of narrative news and the inverted pyramid. Journalism studies, 17(5):608–627.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Marjan Ghazvininejad, Vladimir Karpukhin, and Asli Celikyilmaz. 2021. Discourse-aware prompt design for text generation. arXiv preprint arXiv:2112.05717.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 238–252, Online. Association for Computational Linguistics.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. Transactions of the Association for Computational Linguistics, 6:437–450.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. arXiv preprint arXiv:1904.09751.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In International Conference on Machine Learning, pages 1587–1596. PMLR.

Xinyu Hua and Lu Wang. 2020. PAIR: Planning and iterative refinement in pre-trained transformers for long text generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 781–793, Online. Association for Computational Linguistics.

Zdeněk Kasner and Ondřej Dušek. 2020. Data-to-text generation with iterative text editing. In Proceedings of the 13th International Conference on Natural Language Generation, pages 60–67, Dublin, Ireland. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. arXiv preprint arXiv:1909.05858.

Michael Lebowitz. 1987. Planning stories. In Proceedings of the 9th annual conference of the cognitive science society, pages 234–242.

Leo Leppänen and Hannu Toivonen. 2021. A baseline document planning method for automated journalism. In Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa), pages 101–111, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Robert L Logan IV, Nelson F Liu, Matthew E Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge-graphs for fact-aware language modeling. arXiv preprint arXiv:1906.07241.

Ruqian Lu, Shengluan Hou, Chuanqing Wang, Yu Huang, Chaoqun Fei, and Songmao Zhang. 2019. Attributed Rhetorical Structure Grammar for Domain Text Summarization. arXiv preprint arXiv:1909.00923.

Fuli Luo, Damai Dai, Pengcheng Yang, Tianyu Liu, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. Learning to control the fine-grained sentiment for story ending generation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6020–6026, Florence, Italy. Association for Computational Linguistics.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.

William C Mann. 1984. Discourse structures for text generation. Technical report, UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST.

James Richard Meehan. 1976. The Metanovel: Writing Stories by Computer. Yale University.

Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. Transactions of the Association for Computational Linguistics, 6:373–389.

Archita Pathak and Rohini K Srihari. 2019. Breaking! presenting fake news corpus for automated fact checking. In Proceedings of the 57th annual meeting of the association for computational linguistics: student research workshop, pages 357–362.

Xiangyu Peng, Kaige Xie, Amal Alabdulkarim, Harshith Kayam, Samihan Dani, and Mark O Riedl. 2021. Guiding neural story generation with reader models. arXiv preprint arXiv:2112.08596.

Horst Pöttker. 2003. News and its communicative quality: the inverted pyramid—when and why did it appear? Journalism Studies, 4(4):501–511.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. Plotmachines: Outline-conditioned generation with dynamic plot state tracking. arXiv preprint arXiv:2004.14967.

Alexis Ross, Ana Marasović, and Matthew Peters. 2021. Explaining NLP models via minimal contrastive editing (MiCE). In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3840–3852, Online. Association for Computational Linguistics.

Evan Sandhaus. 2008. The new york times annotated corpus. Linguistic Data Consortium, Philadelphia, 6(12):e26752.

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D Manning. 2019. Do massively pretrained language models make better storytellers? arXiv preprint arXiv:1909.10705.

Alexander Spangher, Jonathan May, Sz-Rung Shiang, and Lingjia Deng. 2021. Multitask semi-supervised learning for class-imbalanced discourse classification. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 498–517.

Alexander Spangher, Gireeja Ranade, Besmira Nushi, Adam Fourney, and Eric Horvitz. 2020. Characterizing search-engine traffic to internet research agency web properties. In Proceedings of The Web Conference 2020, pages 2253–2263.

Felix Stahlberg, James Cross, and Veselin Stoyanov. Simple fusion: Return of the language model.

Manfred Stede and Carla Umbach. 1998. Dimlex: A lexicon of discourse markers for text generation and understanding. In 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2, pages 1238–1242.

Miglena M Sternadori and Kevin Wise. 2010. Men and women read news differently. Journal of media psychology.

Teun A Van Dijk. 2013. News as Discourse. Routledge.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

Jingjing Xu, Xuancheng Ren, Yi Zhang, Qi Zeng, Xiaoyan Cai, and Xu Sun. 2018. A skeleton-based model for promoting coherence among sentences in narrative story generation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4306–4315, Brussels, Belgium. Association for Computational Linguistics.

Kevin Yang and Dan Klein. 2021. Fudge: Controlled text generation with future discriminators. arXiv preprint arXiv:2104.05218.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 7378–7385.

Xinyi Zhou, Atishay Jain, Vir V Phoha, and Reza Zafarani. 2020. Fake News Early Detection: A Theory-Driven Model. Digital Threats: Research and Practice, 1(2):1–25.

A Further Implementation Details

A.1 Discrimintor Implementation

We tested 122 different discriminator variations. A summary of the major architectural iterations is shown in Table 3. We describe each variation as follows; the top-performing variation, with a subset of input sentences and labels, is shown in Figure 7.

Contextualized word vectors (w) from a PTLM (we experimented with either GPT2 or a RoBERTa as in Spangher et al. (2021)) are obtained for each sentence, and are combined using self-attention. Switching to GPT2 yielded a 16-point F1-macro score drop. Hidden-State Control, based on Dathathri et al. (2019), relies on perturbations to the state variable H from the naive language model to generate word-probabilities $p(x_i|X_{<k}, x_{<i}, c) = p(x_i|H, c) = p(c|H, x_i)p(x_i|H)$. So, we need to use the same PTLM for the language model as we do for the discriminator. We do not have the same restriction on Direct Probability Control (Yang and Klein, 2021), as the probabilities are directly multiplied and thus do not need to share any architectural components. For the sake of an apples-to-apples comparison on the mechanism of control, though, we use a GPT2 model for the PTLM layer in our discriminator.

Next, we tested either embedding each sentence separately in batch, or embedding the entire document (+Flattened Sentences). Embedding the entire document yielded a 3 point F1-macro increase.

These sentence vectors are then contextualized: we tested an LSTM layer (+LSTM) to contextualize these vectors and an autoregressive transformer layer (+Transformer)[20]. Using transformer yielded a 6 point F1-macro increase. We next fine-tuned the GPT2 LM using it's LM head on an unlabeled, 30K article news corpus. This yielded a 3 point F1-macro increase.

To incorporate label information as input to the model (as in the Past and Full variants) we embed each label using a learned embeddings layer, and then we combine these embeddings using self-attention[21]. Experimenting with a different window size yielded a 5 point F1-macro increase. We find that a window of 3 yields the best-performing discriminator.

Figure 7: Sentence classification model for $k = 3$ of a 3 sentence document. Word embeddings (w_k) for each sentence (X_k) are combined with self attention (s_k). A transformer contextualizes s_k (a_k) with $s_{<k}$. Labels c are embedded (e) and self-attention generates label vectors (h_k). a_k, h_k are combined for predictions (p).

Discriminator Version	F1 Macro
RoBERTa Baseline	0.62
GPT2	
+ Contextualizing Layer	
LSTM	0.46
Transformer	0.52
+ Flattened Sentences	0.55
+ LM Fine-Tuned with News Corpus	0.58
+ Labels	
Full	0.58
Window=7	0.61
Window=5	0.62
Window=3	0.63
Window=2	0.62

Table 3: F1 Macro on main prediction head, p_k, for different discriminator variations. RoBERTa baseline is from Spangher et al. (2021). GPT2 variations described in body.

Finally, a feed forward classifier combines the sentence vector with the label vector. We find that sharing the PTLM improves accuracy, but not other layers.

A.2 Details on Hyperparameters

A.2.1 Discount Factor, b

To impose further structural control, we impose a prior on t_3 that acts as a discount factor. In words, we downweight the discriminator probabilities for control codes that are farther away from the current sentence being generated. The form of our prior is: $t_3 = \prod_{j=1}^{S} m(i,j)p(c_j|x_i, x_{<i}, X_{<k}, c_{<j})$, where $m(i,j) = b^{|i-j|}$. We experiment with $b = [.33, .66, 1]$. So, the lower the discount factor, b, the more the current, local control code matters. When $b = 0$, the *local-only* variant of our discriminator, Equation 3, is expressed by default.

[20] With 2 layers and 2 attention heads

[21] This architecture allows us to capture structural dependencies between labels better than approaches like a CRF layer, which cannot easily be extended beyond linear-chain operations.

(a) (b)

(c) (d)

Figure 8: The effect of Discount Factor, b, across different pipelines and hyperparameters.

We see in Figure 8 that discount factor b has a non-linear effect on the output. In accordance with our prior results, $b = 0$ is the lowest-performing variant across all four human-quality metrics. $b = .33$ seems to be the most effective discount factor overall, and yields the best output for accuracy and logical flow, while $b = 1$ yield the best-performing output for grammar and topicality. We conclude that a finer-grained balance of local control and structural control might be important overall, but in some cases more structural control might help, as noted previously.

A.2.2 Hidden-State Control (HS)

In Dathathri et al. (2019), authors find anywhere between 3 and 10 backpropogation steps is acceptable. In this work, we use 10 steps with a small step size. We also test different regularizations, also explored in (Dathathri et al., 2019), on the output logits generated from $\hat{H}$. We experiment with different hyperparameters for one of the regularizations: $\hat{l} = \lambda \hat{l} + (1-\lambda) l'$ where l' is the naive, unperturbed logits. We experiment with different values of λ from 0 (fully unperturbed) to 1 (fully perturbed).

A.2.3 Direct-Probability Control (DPC)

Authors in (Yang and Klein, 2021) offer an innovation by training their classifier $p(c|x)$ to consider subsequences $p(c|x_1, ..., x_i)$ for all i, ostensibly improving the accuracy of their joint probability calculation while midsequence. This is in contrast to Dathathri et al. (2019)'s training regimine, which only considers full sequences $p(c|x_1, ..., x_n)$. However, Yang and Klein (2021) do not provide ablations to show whether it is this training regimine, or their direct calculation of $p(x)p(c|x)$, which is responsible for the improvements they observe. In this work, we perform this ablation and find that it has negligible difference, according to automatic evaluation metrics. We also introduce a mean fusion (Stahlberg et al.) into the $p(x)p(c|x)$ joint likelihood: $\lambda p(c|x) + (1-\lambda) p(x)$ and test different values of λ.

B Automatic Metrics List

Here, we discuss the automated metrics reported in Table 2. They are largely based off metrics proposed in See et al. (2019).

B.1 Metrics Reported in Paper

Label Probability: We measure the label probability assigned to the gold-truth class label given in our input sequence $p(c|c_{<s}; x_i; x_{<i}; X_{<s})$. We use head h, or the current head, in the discriminator shown in Figure 7.

Perplexity: Perplexity is calculated using the fine-tuned GPT2 model, which we fine-tuned on 30,000 news articles.

Diverse N-grams: We measure the likelihood that an n-gram in one sentence will be unique compared with the entire document. In other words:

$$\text{Diverse N-Grams}(s; d) = \frac{\#\text{ unique n-grams in sentence } s}{\#\text{ n-grams in document } d} \quad (6)$$

We calculate the set of n-grams per document as the total number of 1,2,3-grams in that document. We calculate one measurement per sentence in the document, and average these scores together.

Sentence Length: We measure the total number of words in the sentence, based on word-level tokenization using https://spacy.io/ .

Unseen Words: We use an external corpus of 30,000 news articles to determine a typical, large news vocabulary. Any words that are outside of this vocabulary are considered "Unseen Words". For our purposes, we are most interested in exploring malformed words, which are sometimes generated by the language model. However, unseen words might also be proper nouns.

C Generation-Baseline #1: Prompting. Further Details

As a baseline, we train a language model to directly calculate $p(x_i|x_{<i}; X_{<s}; c)$, following (Keskar

et al., 2019). We design the following prompt structure to simulate baseline, past-aware and full-sequence control variants.

Baseline:

`Headline: <Headline> Labels: <Current Label> Sentences: <Sentence 1> <Sentence 2>... <Sentence s>`

Past-Aware:

`Headline: <Headline> Labels: <Label 1>, <Label 2> ... <Label k> Sentences: <Sentence 1> <Sentence 2>... <Sentence s>`

Full-Sequence:

`Headline: <Headline> Labels: <Label 1>, <Label 2> ... <Label s> Current Position: <i> Sentences: <Sentence 1> <Sentence 2>... <Sentence s>`

The prompts are specific to current sentence being generated. We first start by generating sentence 1, whereby the prompt for *Baseline* and *Past-Aware* is both:

`Headline: <Headline> Labels: <Label 1> Sentences:`

Then, we let the model generate the first sentence and stop when we generate the < EOS > character. We then regenerate the prompt to include the previously generated sentence and update the tags, so *Baseline* becomes:

`Headline: <Headline> Labels: <Label 2> Sentences: <Sentence 1>`

and *Past-Aware* becomes:

`Headline: <Headline> Labels: <Label 1> <Label 2> Sentences: <Sentence 1>`

We continue in this fashion, resetting the prompt each time, until we have finished generating sentences for all the tags in our input data.

The Full-Sequence process is very similar, except we do not need to update the label-space, since by default the model is exposed to the full sequence of tags before generation.

D Editing

In this section, we describe the various components of the editing model. First, we note the differences in our approach and Ross et al. (2021)'s method. Then, we discuss the infilling model and the discriminator.

D.1 Key Differences

Ross et al. (2021) designed their editor to flip classifier predictions. So, they edited input $x \rightarrow \hat{x}$ until $\max_c p(c|\hat{x}) \neq \max_c p(c|x)$. Then, $(x, \hat{x})$ was given as the explanation for the flip. We are not concerned with flipping predictions so much as maximizing the probability of the ground truth label. So, we design our objective to be $x \rightarrow \hat{x}$ until $p(c|\hat{x}) > p(c|x)$.

To understand why the loss-gradient on the input can provide feature importance, consider the first-order Taylor approximation of the loss, $l(x) \approx l(a) + l'(a)(x - a)$. Here, the gradient of the loss at a, $l'(a)$, can be seen as a set of linear weights similar to logistic regression coefficients, which are commonly used for feature importance.

We also wished to restrict editing to explicit discourse markers, spuriously correlated words, so we heuristically excluded all Proper Nouns, Named Entities (except DATE) or adjectives from the edit candidate set. Table 6 shows explicit discourse markers in the news discourse context. Here, we show the top words associated with each discourse class[22]. Some words effect the tense of the sentence[23], others inject epistemological uncertainty[24], still others time-peg events to certain days[25].

D.2 Infilling Model

We train a label-aware infilling model in a similar method as Ross et al. (2021). Our prompt is:

`label: <label> text: Lorem Ipsum <mask> Lorem <mask> Ipsum.`

Where the masks replace high-salience words, which we discovered as described above. We format samples using sentences in our training dataset, and train a T5 model as described by the authors.

D.3 Possible Improvements

We note that this infilling method directly models $p(\hat{x}|M(x); c)$, i.e., the likelihood of infilled words given a label and a masked sentence. Another possible approach to this problem would be to use a naive infiller and Bayes rule as done in the generation phase of this paper to generate logits $p(\hat{x}|M(x))\,p(c|\hat{x}; M(x))$. This could possibly improve the editor for the same reasons Dathathri

[22] Most positive coefficients of a Logistic Regression Classifier that takes as input a sentence and predicts it's discourse class

[23] Top verbs in *Expectation* are almost all present-tense, while top verbs in *Previous Event* are almost all past-tense

[24] Top verbs in *Evaluation* are all "say" verbs, while verbs in *Current Context* are based on observable events

[25] Top *Main Event* nouns are nearly all weekday names

Discourse Tag	Pre-editing	Post-editing
Consequence	The company has already spent $ 23 billion in Medicare, seeking antitrust clearance.	The company also plans to buy $ 23 billion in Medicare, seeking antitrust clearance.
Expectation	Volvo Car dropped in the first quarter after a trade row over Chinese car makers.	Volvo Car is expected to close lower in the first quarter after a trade row over Chinese car makers.
Evaluation	The deal values Wind Energy, which has operations offshore in New York.	The deal is significant for Wind Energy, which has operations mostly in New York.
Current Context	8 billion shares sold in all of 2015.	8 billion shares were traded in all of China.
Expectation	The deal comes as insurers and drugmakers struggle with competition from Medicare prescription drugs.	The deal could stall as insurers and drugmakers struggle with competition for Medicare prescription drugs.

Table 4: A selection of sentences and the edit operations performed on them. The editor focuses on (a) temporal relations, (b) conditional statements (c) explicit discourse markers (e.g. "expect") and correct grammar.

et al. (2019) and Yang and Klein (2021) observed an improvement over CTRL (Keskar et al., 2019).

Another aspect of the editor that we noticed was that it could sometimes degrade the coherency and topicality of the document. This is especially evident in the **Human** trials. We partially addressed this by selecting candidate edits based off the perplexity of the whole document. We could have mitigated this further by giving our infiller the entire document as context [26].

E Further Methods Comparison

The standard controlled text generation setup is typically expressed as follows:

$$p(x|c) = \prod_{i=1} p(x_i|x_{<i}; c) \qquad (7)$$

where x is the output sequence and c is a single control code (for example: sentiment (Dathathri et al., 2019)). Here, x is a single sentence (or paragraph) of n words, factorized autoregressively into words x_i and previous words $x_{<i}$.

Previous approaches to controlled text generation (Dathathri et al., 2019; Yang and Klein, 2021) factorize the right term of Equation 7 as follows:

$$p(x_i|x_{<i}; c) \propto p(x_i|x_{<i})p(c|x_i; x_{<i}) \qquad (8)$$

As in Equation 7, this factorization decomposes our sequentially controlled text generation model into an uncontrolled language model and a control-code model. The key difference between Equation 8 and 2 is in the second term, i.e. how we choose to model the control codes (the difference in the first term is simply a rather trivial extension of a naive

[26] I.e. we could have trained a model based on $p(\hat{x}|M(x); X_{<s}; c)$, instead of $p(\hat{x}|M(x); c)$

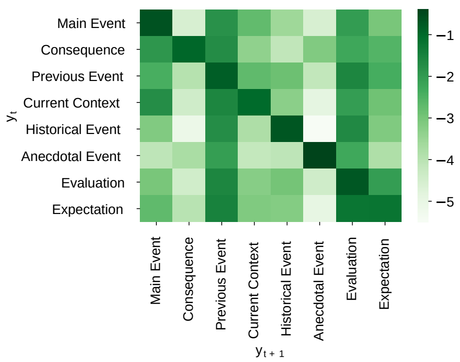


Figure 9: Transition Probability Matrix (log likelihood) for tag sequences.

language from a sentence-to-paragraph generation to a document-generation context).

We show a direct comparison of all of our generation approaches in Figure 5. Here, we show that Direct Probability Control has the best effect over Naive GPT2 for class-accuracy and, surprisingly, perhaps, Grammar and Topicality as well.

F Ovid's Unicorn Is Not Structural

We annotate of the famous Ovid's Unicorn news article generated and presented by the original GPT2 authors (Radford et al., 2019).

We analyse this article as we have analyzed our generation models Section 7. One of our annotators gave each sentence the Van Dijk discourse label that best fits (Van Dijk, 2013), and the other assessed whether it actually fit. This is not an apples-to-apples comparison with the **Label Acc.** column in Table 2, because we are assessing the accuracy of the label that *we* chose *after* reading the text.

We next measured the likelihood that an article with the discourse structure of Ovid's Unicorn would exist naturally. We build a simple bigram model for tags, $p(c_{t+1}|c)$, to calculate the total

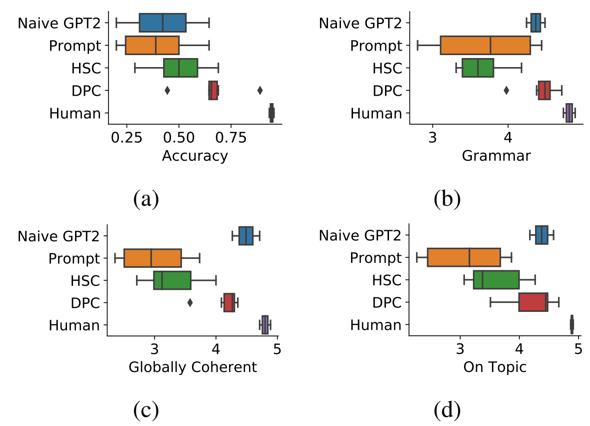


Figure 10: Different generation methods, across different pipelines and hyperparameters.

Article Source	Average Log-Likelihood
Test Set (5/50/95 Percentile)	-1.28/-1.60/-2.01
Ovid Unicorn's	-2.24

Table 5: Log-Likelihood of Tag-Sequence, according to simple bi-gram model $p(c_{t+1} j c_t)$, trained by counting tag sequences in the training dataset. 5th/50th/95th percentiles shown for test set.

probability of a tag sequence.We show in Figure 9, the typical transitions between discourse labels in the news discourse dataset. We fit our simple bigram model using label sequences in the training dataset, and calculate average log-likelihood of the tag sequence for each document in our test dataset. The median of across these is shown in Table 5. As can be seen, sequences in the test dataset are far more likely than the Ovid's unicorn article, which falls outside of the 95th percentile of the distribution of typical articles.

F.1 Van Discourse-based Schema Introduced in Choubey et al. (2020)

The schema used for *News Discourse*, introduced by (Choubey et al., 2020), was based off the schema introduced by Van Dijk (2013). As such, the classification guidelines were:

Main Event : The major subject of the news report. It can be the most recent event that gave rise to the news report, or, in the case of an analytical news report, it can be a general phenomenon, a projected event, or a subject.

Consequence : An event or phenomenon that is caused by the main event or that directly succeeds the main event.

Discourse Label	Top words		
Main Event	monday	cooperation	shot
Consequence	closed	showed	issued
Previous Event	comment	declined	agency
Current Context	shot	prime	groups
Historical Event	2015	2016	2017
Anecdotal Event	want	told	old
Evaluation	say	think	told
Expectation	expected	likely	continue

Table 6: Top predictive words for each discourse type (top positive coefficients for a Logistic Regression trained to predict y = news discourse tag per sentence using and X = a bag of words representation of each sentence).

Previous Event : A specific event that occurred shortly before the main event. It either directly caused the main event, or provides context and understanding for the main event.

Current Context : The general context or world-state immediately preceding the main event, to help the readers better understand and contextualize the main event. Similar to **Previous Event**, but not necessarily tied to a specific event.

Historical Event : An event occurring more than 2 weeks prior to the main event. Might still impact or cause the main event, but is more distal.

Expectation : An analytical insight into future consequences or projections made by the journalist.

Evaluation : A summary, opinion or comment made by the journalist on any of the other discourse components.

Anecdotal Event : Sentences describing events that are anecdotal, such events may happen before or after main events. Anecdotal events are specific events with specific participants. They may be uncertain and can't be verified. A primary purpose of this discourse role is to provide more emotional resonance to the main event.

In Table 6 we attempt to provide more insight into different News Discourse elements by modeling using Logistic Regression.

G Annotation

We recruit two manual annotators, one with > 1 year and the other with > 4 years of journalism experience. Both annotators offered to perform these tasks voluntarily in exchange for acknowledgement.