Sequentially Controlled Text Generation

Anonymous ACL submission

Abstract

While GPT2 generates sentences that are remarkably human-like, longer documents can ramble and are structurally different from human-written articles. 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, based on extensions of existing classifier-based approaches. We test different degrees of structural awareness and show that, in general, more structural awareness results in higher control-accuracy, grammaticality, global coherency and topicality, approaching human-level writing performance.

1 Introduction

Imagine that you are tasked with "Write a *Related Works* section". Would it help to know that it is coming after the *Discussion* section, or after the *Introduction* but before the *Problem Statement*? In this work, we study the role of structural awareness in narrative text generation.

Numerous works have shown that keyword-planning (Yao et al., 2019), plot-design (Rashkin et al., 2020) and entity tracking (Peng et al., 2021) improves the coherence of generated narratives. However, controlling macro-structural elements, like discourse roles, is underexplored. This is despite well-established findings that natural writing follows macro-structures (Po¨ ttker, 2003; Van Dijk, 2013), which aid in human understanding (Emde et al., 2016; Sternadori and Wise, 2010) and automatic tasks like event extraction (Choubey et al., 2020), summarization (Lu et al., 2019; Isonuma et al., 2019), and misinformation detection (Abbas, 2020; Zhou et al., 2020).

Naive language models, however, generate text that is structurally dissimilar to human-written text (Section 7), even when fine-tuned on in-domain corpora. We show that even the well-known Ovid's Unicorn generation, which seems like a natural news article, exhibits an atypical discourse pattern (Appendix F).

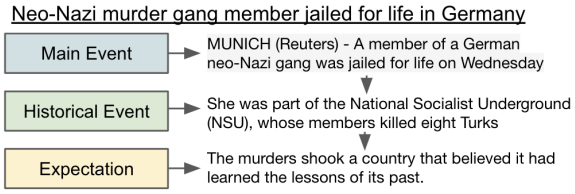
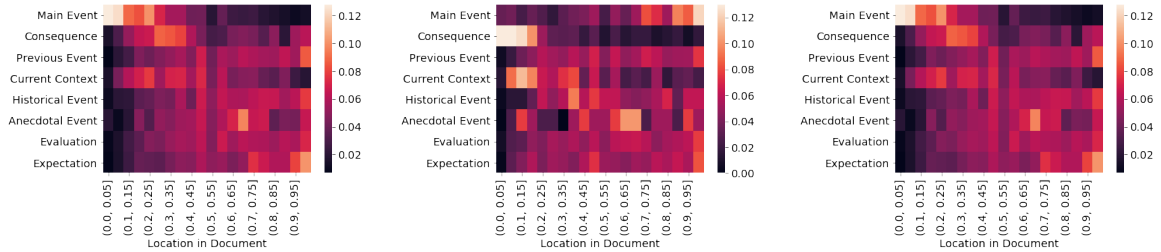


Figure 1: We study the task of *sequentially-controlled generation*: generating documents exhibiting structure given by a sequence of local control codes. Shown is a news article with it's Van Dijk structure (Van Dijk, 2013) and headline. Our models take as input the headline and discourse tags and generate 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.

In the current work, we propose an approach to controlling the macro-structure of texts. We propose a novel task, *sequentially controlled text generation*. In our task, the user provides a sequence of local control codes, each guiding the generation of a sentence. We build on prior work where a single control code was used to generate for multiple sentences (Keskar et al., 2019; Dathathri et al., 2019; Yang and Klein, 2021).

First, we develop and compare novel approaches for generating sequentially controlled text using headlines as prompts and news discourse tags (Van Dijk, 2013) as local control tags, as shown in Figure 1. Next, we test what degree of structural awareness yields the highest-quality documents: **local** (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



(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.

full-sequence (where the generator is aware of the entire document's sequence of control codes). We show that, overall, full-sequence control generates the highest-quality text over a range of metrics.

Finally, we show how to combine *structural* and *local* control in a pipeline. First, we generate a text sentence using structural methods described above, and then, we selectively edit that sentence, using a sentence-level editing technique. When using both techniques in tandem we are able to generate fluent documents that exhibit appropriate structure.

In summary, our core contributions are:

- We propose a novel task, *sequentially controlled text generation*, identify an appropriate discourse schema to represent structure in news articles (Van Dijk, 2013). We use a dataset annotated by Choubey et al. (2020) based on this schema to explore this task.
- We explore three different structural control approaches: *local*, *past-aware* and *full-sequence* control. We show that overall, *full-sequence* produces the highest-quality text over an array of metrics.
- 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.

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) and stylistic generation (He 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 (i.e. "Previous Event" discourse), to aid efforts in computational journalism (Cohen et al., 2011).

2 Problem Statement

We assume, as input, a headline sentence, X_0, and a sequence of control codes $\vec{c} = c_1, c_2, ...c_S$ of length S (i.e. one for each sentence we wish to generate in the document). We wish to produce, as output, a document $\mathbf{X}$ of length S as a sequence of sentences $\mathbf{X} = X_1, ...X_S$, 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:

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

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 $\vec{c}$, the entire sequence of control-codes for a document, is given.

We use Bayes rule to factorize t_1 into:

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

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 training a model scratch.

Three approximations for t_3 are:

Baseline $t_3 \approx p(c_s|x_i, x_{<i}, X_{<s})$ (3)

In the baseline 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.

Past-Aware

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

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

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

In the full-sequence model, we make no conditional independence assumptions. $\vec{c}$ is dependent on x_i in t_3 and x_i is dependent on $\vec{c}$ in t_1.

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 $p(\vec{c})$ to induce a discount factor[2]. This focuses the generator on control code c_k and down-weights surrounding control codes.

In the next sections, we will show how we model these objectives. We start in Section 3 by describing the dataset and schema we use for local control. In Section 4, we describe the discriminator we use as our control-code model, the controlled generation techniques and the editing techniques we adapt.

3 Datasets and Schema

The form of local 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[3], tagged on the sentence level. Their schema consists of 9 classes: { Main Event, Consequence, Current Context, Previous Event, Historical Event, Anecdotal Event, Evaluation, Expectation }.[4]. 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[5] 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.

4 Methodology

We hypothesize that, in practice, introducing local control (Equation 3) *and* structural control (Equations 4, 5) is important for modeling Equation 1. So, we design a pipeline to provide both of these forms of control. We achieve this by oscillating between providing *structural control* during **Generation** and *local control* during **Editing**. The flow we use to accomplish this is depicted in Figure 3.

The first row is the **Generation** row. Here, we impose *structural control*, or approximate Equation 1 to guide our sampling of each word, x_i. When we have completed a sentence, we move to the second row, **Editing**. Here, we edit the sentence to further impose *local control* on each sentence, updating x to optimize a variation of Equation 1: $p(x_i|x_{-i}, c_k)$.

As described in Section 2 (Equation 2), we can efficiently do generation by combining a naively-trained language model with a discriminator (Dathathri et al., 2019). We start by describing the discriminator in Section 4.1 and how it models different types of structural awareness (Equations 3, 4 and 5). Next, we move on to the generation techniques that allow us to combine the discriminator with the naive language model in Section 4.2. Finally, we close by discussing the Editing component in Section 4.3.

4.1 Discriminator

As mentioned above, the discriminator is separately-trained component of our generation pipeline used to generate class-conditional text. The discriminator we construct takes as input a sequence of sentences ($\mathbf{X}$) and a sequence of local control tags ($\vec{c}$). Our architecture combines a

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

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

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

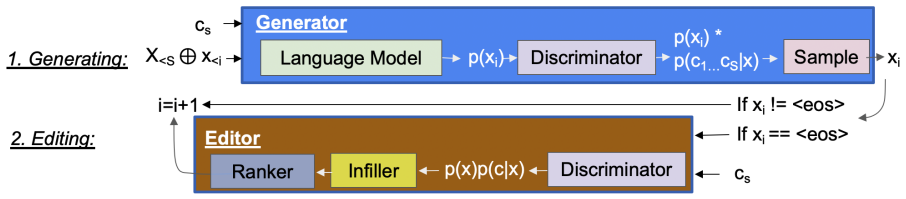


Figure 3: **Generation process**. First, we perturb the output of a language model using a structurally-aware classifier to approximate $p(x_i|x_{<i}, X_{<k})p(\vec{c}|x_{<i}, X_{<k})$ and generate word x_i by sampling from the perturbed distribution . When we generate an $< eos >$ token, we edit the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

sentence-classification model, similar to that used in Spangher et al. (2021), with separate a label embedding architecture (to incorporate $c_{<j}$, X_k and $x_{<i}$ into the conditional of Equation 2, t_3). For a full description of architecture, see Appendix A.

We train it to model Baseline, Past and Future control variants expressed in Section 2 (Equations 3, 4 and 5) in a multi-headed fashion: 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 **Baseline** control (Equation 3) we only use predictions from the main head, k. In **Past Aware** control (Equation 4), we multiply predictions from heads prior to the current sentence $< k$, and for **Full-Sequence** control, we multiply predictions from all heads.[6] We now describe how we use these predictions.

4.2 Generation

We are now ready to describe how we combine our discriminator's class predictions with a naive PTLM to solve Equation 2. We compare two controlled generation methods: **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, resulting 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, following Dathathri et al. (2019). 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. DPC has the advantage of being simpler to implement and more 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.

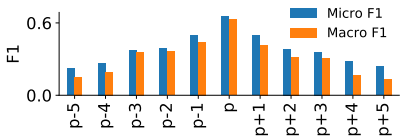


Figure 4: **Discriminator performance** on test data. F1 scores for $p(c_j|X_{<k}, x_{<i}, c_{<j})$ predictions. Sentence index k and word index i are fixed: we show error for using the current sentence to predict all past, current and future labels.

4.3 Editing

Our goal, after we have finished generating a sentence via aforementioned methods, is to make another pass and introduce even more discourse markers to further increase its control-code likelihood.

Following Ross et al. (2021), we use the loss-gradient on our input sentence X_k to identify words with the highest gradient (i.e. they have the most impact on class predictions). These are our candidates for masking. We further cull this set based on heuristics[7] to encourage the editor to introduce explicit discourse markers rather than topic words spuriously correlated with labels.

Then, we fine-tune a label-aware infilling model (Raffel et al., 2019) to generate candidate edits[8]

[6]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.

[7]Words that are *not* proper nouns, named entities (except the DATE class) or adjectives.

[8]The infilling model, a T5 model, is trained using a labeled training dataset with a specific input template

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.<|endoftext|>
Previous Event	The world's largest retailer of meat and fish products hopes to become one day the leader in convenience groceries.<|endoftext|>
Expectation	It says it could become a leader in fresh fruits and vegetables, as well.<|endoftext|>
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.<|endoftext|>
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.<|endoftext|>
Evaluation	"What's at stake here is transparency, a sense of fairness to all the stakeholders.<|endoftext|>
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.<|endoftext|>
Expectation	He says there are potential conflicts of interest, because one partner is the government.<|endoftext|>
Expectation	"What if you're a government contractor?"<|endoftext|>

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

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 of fluency and class likelihood. We calculate fluency using perplexity of the entire generated document so far, $PPL(x_k \oplus X_{<k})$, to encourage edits preserving the logical flow of the document.

5 Implementation Details

We fine-tune a `GPT2-base` model on a large news corpora with a max word-piece length=2048[9]. 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[10]. However, performance degrades rapidly for heads farther from p.

6 Experiments

We test different pipeline settings with data from the test set of our discourse dataset ($n = 200$). The input to our models, as stated previously, is **a headline (as a prompt) and the *full sequence of gold-truth discourse labels*** from that document. In addition to the pipelines discussed in Section 4, we also compare a baseline method (**Prompting**), **Naive GPT2** generation (given only the headline as input) and **Human**-written articles.

For **Naive GPT2**, we simply generate text using a headline (i.e. no control codes). For **Prompting**, we directly train a class-conditional language model to generate text. We directly model t_1 by including labels in the prompt, as in Keskar et al. (2019). Baseline control is achieved by only including the local control code in the prompt. For Past-Aware control, we include all control codes prior to our current sentence in the prompt. Finally, for Full-Sequence Control, we including the full sequence of control codes in the prompt. (See Appendix C for more details and examples of prompt design.) Finally, for **Human**, we include the *original* article text of the test documents as additional ***blind*** trials in our manual evaluation. For each of these baselines, we test with and without editing (with the human-written text in **Human** and with the generated text in all other trials).

For all pipelines, we select the best hyperparameter configurations based on perplexity and model-assigned class likelihood. Then, we manually annotate each generated document for 4 metrics: **Accuracy (0-1)**[11] **Grammar (1-5)**[12], **Global Coher-**

incorporating the label. An example templated sentence: `label: Background. text: The senator <MASK> to the courtroom to <MASK>.`

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

[10] .71 Micro-F1 in Spangher et al. (2021), which used auxiliary datasets.

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

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

ence (1-5)[13] and **Topicality (1-5)**[14]. We recruit two expert annotators with journalism experience to perform annotations blindly without awareness to which generation pipeline was used, and find moderate agreement $\kappa \in [.3, .5]$ 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.

7 Results

Best Overall Trial We show automatic and human metrics on a subset of trials in Table 3. The highest-performing generation pipelines, across our four human metrics, are all variations of DPC with past or full structural control. We observe that the DPC-Generation, with Past-Aware Control and Editing has the highest class-label accuracy, nearly approaching the human trials. The top-performing pipeline for Global Coherence is also DPC-Generation with Past-Aware Control, but without editing. And the top performing pipelines for grammar and topicality are also DPC-Generation pipelines, but with Full-Sequence Control and without editing.

Effect of Different Pipeline Components We show the effects of each experimental method by showing the distributional shifts in performance across all trials, in Figures 5, 6, 7.

In terms of generation approaches, we show in Figure 5 that Direct Probability Control has the most positive effect over Naive GPT2 for class-accuracy and, surprisingly, perhaps, Grammar and Topicality as well. Structural control has a largely positive effect on generated text. In Figure 5 , we find that Full-Sequence models are, on average, able to generate the most label-accurate sentences with the best grammar, global coherence and topicality. Finally, editing improves Accuracy, Grammar and Global Coherence (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 3, editing can *also* be applied to human-written text to boost label accuracy, but at the slight expense of our coherence metrics.

[13]Global Coherence: how well a sentence functions in the flow of the story

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

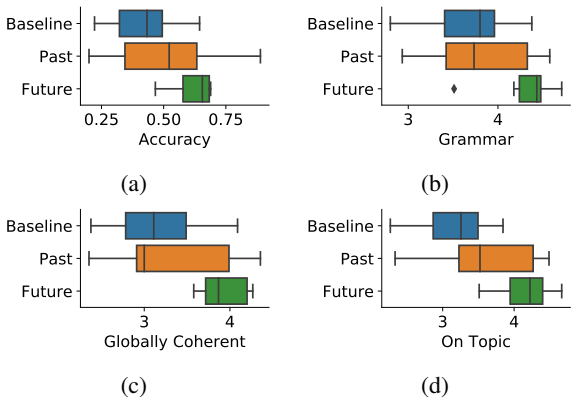


Figure 5: Comparison of different structural control methods across different trials and hyperparameters.

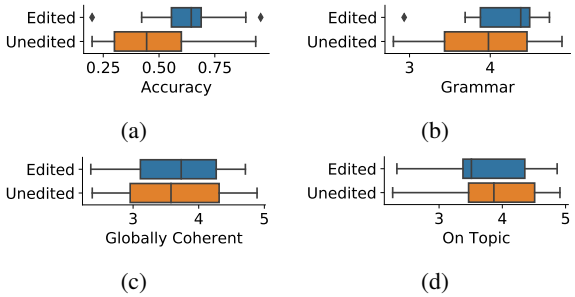


Figure 6: The effect of editing, across different trials and hyperparameters.

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 (past, full, local) had the greatest effect on discourse, overall coherence, topicality and grammaticality. It seems that, on average, full-sequence control seems to generate text that scores highest in these metrics, but in specific high-performing cases (e.g. DPC), past-aware structure is the most performant. It might be that for this level of performance, more structural awareness equates to more noise for the generator.

Insight #1: Weak classifiers can still impose accurate control. At .61 macro F1, our GPT2-based classifier is a relatively weak classifier. Previous work in classifier-based controlled text generation used large training datasets and classifiers that routinely scored above .8 F1 (Dathathri et al., 2019; Yang and Klein, 2021). The weakness of our classifier is one reason why HSC may have performed poorly, however, in other trials we see quite strong human-evaluated accuracy. Note in human-generation; our annotators blindly scored

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 2: 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.

		Human-Annotated Metrics				Automatic Metrics				
Generation	Structure	Label Acc.	Grammar (1-5)	Global Coher. (1-5)	On-Topic (1-5)	Label Prob. (%)	Perplex.	Diverse Ngrams (%)	Sent. Len.	Unseen Words (%)
Naive GPT2		20.0/64.4	4.2/4.5	4.7/4.3	4.6/4.2	4/7	48.2/45.4	7.1/8.3	24.9/38.8	4.7/3.2
Gen-Base: Prompt	baseline	22.2/51.1	2.8/3.9	2.4/3.0	2.3/2.8	5/7	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	14/4	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	5/4	42.5/49.2	7.3/7.8	35.5/42.6	4.6/4.9
Method #1: HSC	baseline	28.9/42.2	3.3/3.7	2.7/3.2	3.1/3.4	10/12	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	11/8	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	10/9	134.5/129.6	7.2/7.8	17.3/20.7	7.0/7.1
Method #2: DPC	baseline	44.4/64.4	4.0/4.4	3.6/4.1	3.8/3.5	2/22	42.1/39.9	5.8/8.3	24.8/42.6	4.7/3.0
	past	64.4/88.9	4.5/4.6	4.4/4.3	4.4/4.5	13/9	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	11/8	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	6/6	34.2/41.0	8.7/8.7	37.9/39.6	4.2/4.5

Table 3: Metrics on different trial runs. Each cell shows Unedited/Edited variants. Metrics are calculated as the median of 1000 bootstrapped trials. (Hyperparams = $\gamma = .75$, $b = .33$).

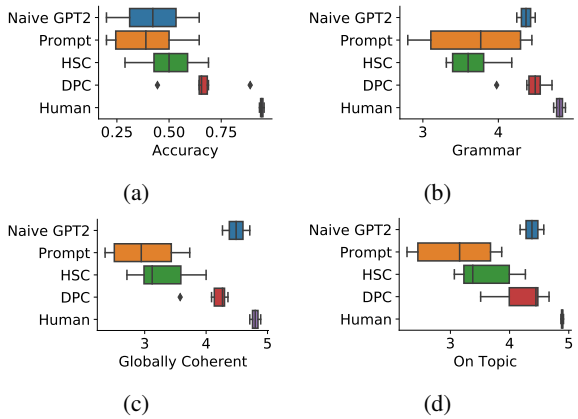


Figure 7: Different generation methods, across different trials and hyperparameters.

these trials above 93%, yet our classifier assigned .06 to the true class. Thus, even with a weak classifier, we can control generation. This might be because the classifier, despite giving low overall probability to the gold-truth tag, can still differentiate between a poor generation from one that better matches the class.

Insight #2: 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 infilling decisions, it would improve class-accuracy but hurt other metrics of document quality, like topicality and coherence. Indeed, for the top-performing trials, like DPC and Human, Editing only improves class accuracy. However, grammar and coherence improves in other trials. This could be because, as mentioned in Section 4.3, 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, and usually occurred after punctuation like decimal points that were misinterpreted by the model to be periods. Indeed, the correlation between sentence-length and grammar is relatively high ($r = .34$). One hypothesis for this is that error-prone sentence tokenizing models provided faulty training data, either during pretrainining or fine-tuning of LMs. This will continue to hinder document-level structural work, which often relies on sentence-boundaries. Another observation, in Table 3, 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 utilizes the MiCE framework (Ross et al., 2021), which is 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[15] 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 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 final 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 [16] 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.

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

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

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 fluency 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 identification based on nlp. *European Journal of Computer Science and Information Technology*, 9(1):1–26.

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.

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.

He He, Nanyun Peng, and Percy Liang. 2019. Pun generation with surprise. *arXiv preprint arXiv:1904.06828*.

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.

Masaru Isonuma, Junichiro Mori, and Ichiro Sakata. 2019. Unsupervised Neural Single-Document Summarization of Reviews via Learning Latent Discourse Structure and its Ranking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2142–2152, Florence, Italy. 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 Po¨ ttker. 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.

Soumya Sanyal and Xiang Ren. 2021. Discretized integrated gradients for explaining language models. *arXiv preprint arXiv:2108.13654*.

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*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

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*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

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

Contextualized word vectors ($\vec{w}$) from a PTLM are obtained for each sentence, and are combined using self-attention. These sentence vectors are then contextualized using a transformer layer[17]. To incorporate label information as input 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[18]. Finally, a feed forward classifier combines the sentence vector with the label vector.

We experiment with sharing different layers in the architecture and find that only sharing the PTLM helps. Because PPLM requires that the input features to the classifier be the same as the hidden features in the language model, we use a frozen GPT2 architecture for to generate contextualized word-embeddings.

Note that Hidden-State Control, based on Dathathri et al. (2019), relies on perturbations to the hidden variable H from the naive language model to generate word-probabilities $p(x_i|X_{<k}, x_{<i}, \vec{c}) = p(x_i|H, \vec{c}) = p(\vec{c}|H, x_i)p(x_i|H)$. So, in practice, we need to use the same PTLM for the language model as we do for the discriminator. Empirically, in early trials, we see a drop in discriminator performance as a result of using GPT2 instead of RoBERTa, as used in Spangher et al. (2021).

Further, note that 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. We could use a different, more classification-designed base than GPT2, but for the sake of an apples-to-apples comparison on the mechanism of control, rather than the discriminator's effect, we use a GPT2 model for the PTLM layer in our discriminator.

A.2 Details on Hyperparameters

A.2.1 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,

[17] With 2 layers and 2 attention heads

[18] 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.

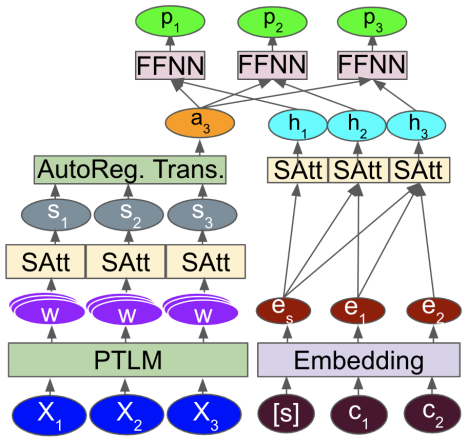


Figure 8: **Sentence classification model** for $k = 3$ of a 3 sentence document. Word embeddings ($\vec{w}_k$) for each sentence (X_k) are combined with self attention (s_k). A transformer contextualizes s_k (a_k) with $s_{<k}$. Labels $\vec{c}$ are embedded (e) and self-attention generates label vectors (h_k). a_k, h_k are combined for predictions ($\vec{p}$).

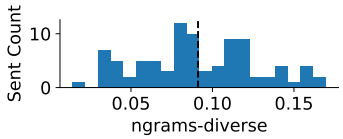
also explored in (Dathathri et al., 2019), on the output logits generated from $\hat{H}$. We experiment with different hyperparameters for one of the regularizations: $\hat{l} = \gamma\hat{l} + (1-\gamma)l^0$ where l^0 is the naive, unperturbed logits. We experiment with different values of γ from 0 (fully unperturbed) to 1 (fully perturbed).

A.2.2 Direct-Probability Control (DPC)

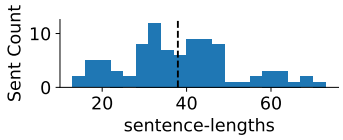
Authors in (Yang and Klein, 2021) offer an innovation by training their classifier $p(c|x)$ to consider subsequences $p(c|x_1, ...x_i)$ for all i, ostensibly improving the accuracy of their joint probability calculation while midsequence. This is in contrast to Dathathri et al. (2019)'s training regimine, 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: $\gamma p(c|x) + (1-\gamma)p(x)$, similarly to Dathathri et al. (2019), and test different values of γ.

B Automatic Metrics List

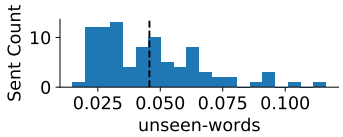
Here, we discuss the automated metrics reported in Table 3. They are largely based off metrics proposed in See et al. (2019).



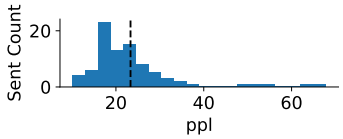
(a) Distribution over diverse n-grams in human articles



(b) Distribution over all sentence Lengths in human articles.



(c) Distribution over all unseen words in human articles.



(d) Distribution over perplexity in human articles.

Figure 9: The automatic metrics we evaluated in our paper follow a natural, Gaussian-like distribution on natural human-written text. Empirical mean shown with dashed line.

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 p, or the current head, in the discriminator shown in Figure 8.

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 to get a single document-level score.

Sentence Length : We measure the total number of words in the sentence, based on word-level tokenization using `https://spacy.io/`, not word-pieces.

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.

B.2 Other Metrics Considered

Figure 10 shows several metrics that we ultimately did not include in the paper because they were correlated with other metrics, did not show much variance, or did not illuminate our problem. For the sake of completeness and for other researchers studying generation problems, we describe them briefly here:

Pred Acc @2 : Number of times the gold-truth tag was in the top 2 tags predicted by our discriminator. This showed more variance and better handled the class-imbalanced nature of our problem than pure Accuracy.

N-Grams Entropy : Entropy measure on sentence-level bag-of-ngrams vectors that we calculated measuring all 1,2,3-grams.

Rare Words : We measured the empirical likelihood of words in a separate news corpora. Rare words is the log likelihood of all words in our generated text according to this empirical likelihood.

C Generation-Baseline #1: Prompting. Further Details

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

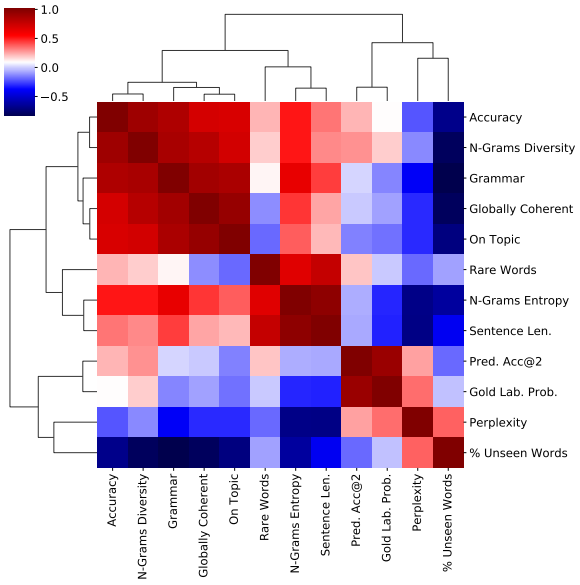


Figure 10: Spearman correlation between different metrics. All automatic metrics are standardized to z-scores.

`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

The biggest difference is that Ross et al. (2021) designed their classifier-aware editor explicitly to flip classifier predictions in order to generate explanations. So, they edited input $x \rightarrow \hat{x}$ until ${}_{c}p(\vec{c}|\hat{x}) \neq_{c} p(\vec{c}|x)$. Then, $\Delta(x, \hat{x})$ was given as the explanation for the flip. On the other hand, 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 $p(c|\hat{x}) > p(c|x)$.

The second key difference is that we further wished to edit in a way that changed *explicit discourse markers*, not topic words that happened to be correlated with the input. To understand what we mean by explicit discourse markers in the news discourse context, see Table 6. Here, we show the top words associated with each discourse class by examining coefficients of a Logistic Regression Classifier that takes as input a sentence and predicts it's discourse class. As can be seen, some of these effect the tense of the sentence (top verbs in *Expectation* are almost all present-tense, while top verbs in *Previous Event* are almost all past-tense). Others inject epistemological uncertainty (top verbs in *Evaluation* are all "say" verbs, while verbs in *Current Context* are based on observable events.) Still others time-peg events to certain days (top *Main Event* nouns are nearly all weekday names). We devise heuristic rules to capture these explicit discourse markers. We exclude all gradients on Proper Nouns, Named Entities (except DATE) or adjectives.

D.2 Infilling Model

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

`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 et al. (2019) and Yang and Klein (2021) observed an improvement over CTRL (Keskar et al., 2019). We considered this but ultimately did not have time to implement it. It's likely, too, that the effect would be more limited then observed for sequential control, because our infilling model is on the sentence-level, rather than the document-level, and thus we have more data that is less sparse.

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 *infilling* based on the entire document. So, instead of generating infills only based on the current sentence, $p(\hat{x}|M(x), c)$, we could have trained a model based on $p(\hat{x}|M(x), X_{<s}, c)$. Time constraints also prevented us from experimenting with this route.

D.4 Background on Gradients

Ross et al. (2021) observe loss-gradients on the input text to find salient words, as in Simonyan et al. (2013). To understand why gradients on the loss can provide feature salience, consider a simple linear model, $Y = wX + b$. w provides a direct model for the effects of each input feature on the output. Now, consider a nonlinear function: $Y = f(X)$. In the first-order Taylor approximation of this model:

$$\begin{aligned} f(X) & \\ &= f(a) + f'(a)(x-a) + h(x)(x-a) \\ &\approx f(a) + f'(a)(x-a) \end{aligned} \quad (7)$$

Here, the gradient $w = f'(a)$ is a linear model at point a that approximates where shifts in X shift the output. So, it serves the same function as w in the linear model. Despite being an approximation, we find it is a good approximation and time-efficient, especially compared with other methods like Integrated Gradients (Sundararajan et al., 2017) or Discretized Integrated Gradients (Sanyal and Ren, 2021). Both of these methods have been shown to better approximate feature-importance than a pointwise gradient, but we found they were prohibitively slow. We note that the original authors also experimented with Integrated Gradients and, via correspondence, told us that they also found them to be too slow.

E Further comparison

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

$$p(x|c) = \prod_{i=1}^{n} p(x_i|x< i, c) \quad (8)$$

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 inner-term of Equation 8 as follows:

$$p(x_i|x_{<i}, c) \propto p(x_i|x_{<i})p(c|x_i, x_{<i}) \quad (9)$$

As in Equation 8, this factorization decomposes our sequentially controlled text generation model into an uncontrolled language model and a control-code model. The key difference between Equation 9 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 language from a sentence-to-paragraph generation to a document-generation context).

S_idx	Sentence	Best Label	Label Acc.	Gram-mar	Global Coh.	On-Topic
0	The scientist named the population, after their distinctive horn, Ovid's Unicorn.	Main Event	1	5	5	5
1	These four-horned, silver-white unicorns were previously unknown to science.	Current Context	1	5	5	5
2	Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.	Evaluation	1	5	4	5
3	Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans.	Previous Event	1	5	3	5
4	Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.	Anecdotal Event	0	5	4	2
5	Pérez and the others then ventured further into the valley.	Previous Event	1	5	5	3
6	"By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.	Anecdotal Event	0	5	4	2
7	Pérez and his friends were astonished to see the unicorn herd.	Anecdotal Event	0	5	3	5
8	These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.	Anecdotal Event	1	3	4	5
9	While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English.	Main Event	1	5	5	5
10	Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."	Evaluation	1	4	5	5
11	Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.	Historical Event	1	5	3	4
12	While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization.	Evaluation	1	5	3	5
13	According to Pérez, "In South America, such incidents seem to be quite common."	Evaluation	1	5	4	4
14	However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA.	Expectation	1	5	2	3
15	"But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.	Evaluation	1	5	5	5
	Ovid's Unicorn Average		81.2	4.8	4.0	4.3
	Naive GPT2 (from trials with gold labels)		20.0	4.2	4.7	4.6
	Our best method		88.9	4.6	4.3	4.5
	Human		93.3	4.9	4.9	4.9

Table 4: A somewhat humorous attempt to show via anecdote that Naive GPT2 output, while appearing at first glance to contain standard news discourse elements, fails to produce articles that resemble the structure of typical news articles. We hand-label the most likely discourse tag for each sentence generated by GPT2, then we calculate the four human metrics calculated in the main body of the paper. (Note that because we assign the label *and* judge whether it fits, this is more a measure of whether the sentence fits *any* Van Dijk news discourse tag (Van Dijk, 2013)).

F Ovid's Unicorn Is Not Structural

We show in Table 4 a manual annotation performed by the researchers of the famous Ovid's Unicorn news article generated and presented by the original GPT2 authors.

We attempt to analyse this article as we have analyzed the output from our generation models in Section 7. In the **Best Label** column, we attempt to assign the discourse label that best fits each sentence, from Van Dijk's schema (Van Dijk, 2013). In the **Label Acc.** column, we assess whether the label we assigned actually fits the sentence itself. This is not an apples-to-apples comparison to the **Label Acc.** column presented in Table 3, because we are assessing the accuracy of the label that *we* chose *after* reading the text, in contrast to Table 3, where we simply assess whether the text matches the gold-truth label that we passed into the model. But the whole purpose of this section is to be pedantic, anyway, so let's go with it. In a sense, what this column represents is whether each sentence in Ovid's Unicorn matches *any* discourse label in Van Dijk's news discourse schema (Van Dijk, 2013). See Appendix F.1 for definitions for each of the discourse labels in the schema.

In the **Grammar** column, we applied stringent standards, looking not just for Grammatical accuracy, but also local coherence: does the sentence make sense interally? Sentences 8 and 10 each contain internal mistakes. In Sentence 8, there is coreferential confusion between the different parties that "they" refers to; in one interpretation, one "they" refers to the researchers and another refers to the "unicorns". It does not make sense for the unicorns to be seen from afar while also being close enough to touch. In Sentence 10, "dialectic" does not make sense in this context. For **Global Coherence** and **Topicality**, we applied the same criterion that we applied to the metrics we calculated in Table 3.

So, we see that Ovid's Unicorn does not have the grammatical quality, coherence, or topicality that structurally controlled output does, it has enough sentences that do not fit any discourse classes so that, even when we both choose the label and evaluate it's goodness-of-fit, Ovid's unicorn still does not score well on label accuracy!

Let us defend our choices where **Label Accuracy** is 0: Sentences 4, 6 and 7. For Sentences 4 and 6, we struggled with whether to define each sentence as an Anecdotal Event, Previous Event or Current Context. According to the guidelines in Appendix F.1, each sentence describes in more detail the event (i.e. "the researchers' journey") that directly preceded the event described in the *Headline* and Main Event sentences (i.e. "discovering the unicorns; discovering that they spoke English"). So, in the sense that this event temporally precedes the Main Event, one might claim that it causes it, and thus is a Previous Event.

However, the primary purpose of these sentences is to describe the setting, which itself does not appear to actually have caused the discovery. They also do not exactly convey a *Current Circumstance*, since simply the presence of a fountain does not help the reader understand the *Main Event* any better. Finally, for lack of a better choice, we saw similarities between this sentence and an *Anecdotal Event* sentence, as it does add color. Still, because (to be pedantic), it does not seem like either sentence truly helps the reader understand the main event better, so we marked *Label Accuracy* = 0 for both.

Sentence 7, similarly, falls somewhere in between *Anecdotal Event* and *Evaluation*. This sentence does add emotional color, similarly to the sentences previously discussed, and so *Anecdotal Event* is an option. However, again, it does not

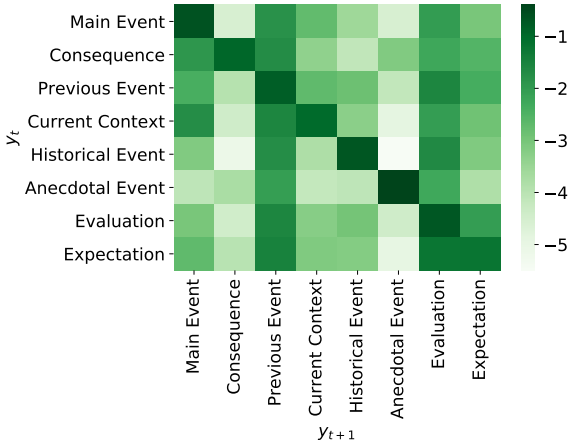


Figure 11: Transition Probability Matrix (log likelihood) for tag sequences.

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 $p(c_{t+1}|c_t)$, trained by counting tag sequences in the training dataset. 5th/50th/95th percentiles shown for test set.

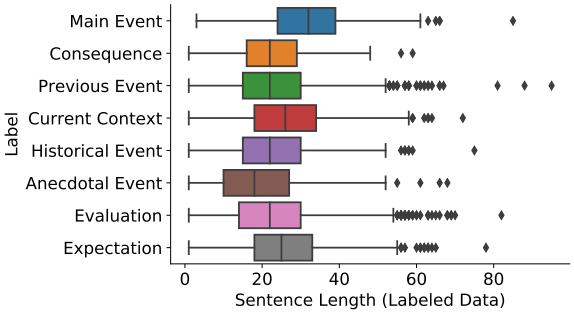


Figure 12: Different discourse categories sometimes have different sentence lengths.

help the reader understand the Main Event. As such, it could be seen as an *Evaluation*, which is a general comment on any other discourse element. This could certainly be argued. We felt that the potential discoursive confusion here warranted that we assign **Label Accuracy** = 0, again.

In all three cases, thus, the discourse function of the sentence could not be precisely articulated, so we were left confused about what the actual purpose of these sentences was. Other researchers have also noted that large language models, which producing high-quality text, can meander (Alabdulkarim et al., 2021; Peng et al., 2021), and that is our main point here.

We close by measuring 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 probability of a tag sequence as follows:

$$p(\vec{c}) \approx \prod_{t=1}^{S-1} p(c_{t+1}|c_t)$$

We show in Figure 11, 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.

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. We show the top most predictive words for each category, ordered by their positive coefficients.

G Annotation

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

Discourse Label	Top β coefficients						
Main Event	monday	cooperation	shot	thursday	statement	wednesday	sunday
Consequence	closed	showed	issued	power	emergency	news	authorities
Previous Event	comment	declined	agency	announced	week	month	quarter
Current Context	shot	prime	groups	march	man	border	data
Historical Event	2015	2016	2017	later	came	ago	shot
Anecdotal Event	want	told	old	declined	said	make	school
Evaluation	say	think	told	interview	added	latest	need
Expectation	expected	likely	continue	plans	face	help	change

Table 6: Explanation into different features and their importance for predicting discourse function. Shown here are top positive β 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.

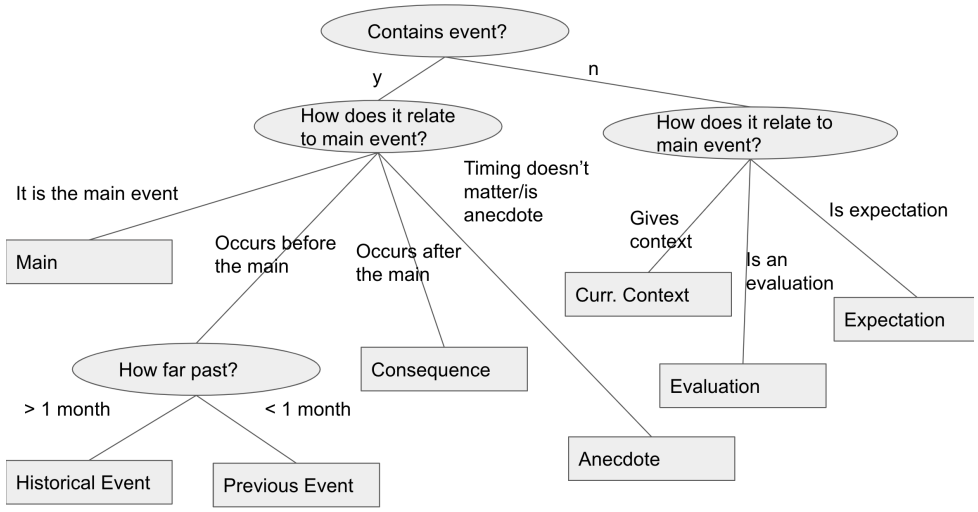


Figure 13: Tree shown to annotators for reference on manual annotation task.

For task instructions, we showed them the label definitions, shown above in Section F.1. We also showed them the decision-tree shown in Figure 13 that we developed for their reference. The decision-tree breaks down key components of discourse reasoning so that annotators can arrive at labels consistent with with guidelines but in more concrete steps.

Additionally, we gave them training annotation questions for practice. For training annotation, they used a modified, interactive version of the decision tree, shown in Figure 14. For the training task, they were asked to view human-written sentences from 10 articles and go through the step-by-step question process that we based on the decision tree (the figure shows an annotator who has already reached the end of the process.) These labels were checked with the gold labels from the training dataset, and they continued training until they were answering questions with >80% accuracy.

We showed them an interactive template, shown in Figure 15 to annotate sentences from the generation models described in our paper. They saw articles randomly, with no knowledge of which trial the models came from. After a few rounds, though, we realized that correctly assigning labels was too difficult, and we were not seeing any meaningful variation in results. So we designed another interface, shown in Figure 16. This interface was similarly randomized, but the quality metrics were on a 5-point scale instead of 3, and it asked annotators to simply agree Y/N if the tag fit or not. This actually gave us a meaningful stratification of classes.

For Grammar, we asked them to count the number of grammar mistakes per sentence. We quantified different grammar-mistake bands (`1: >6, 3:2-4, 5:0`). For Global Coherence, we used a qualititative metric (`1: "Poor", 3: "OK", 5: "Great"`). For Topicality, we also used a qualitative metric (`1: "Not at all", 3: "OK", 5: "Great"`)

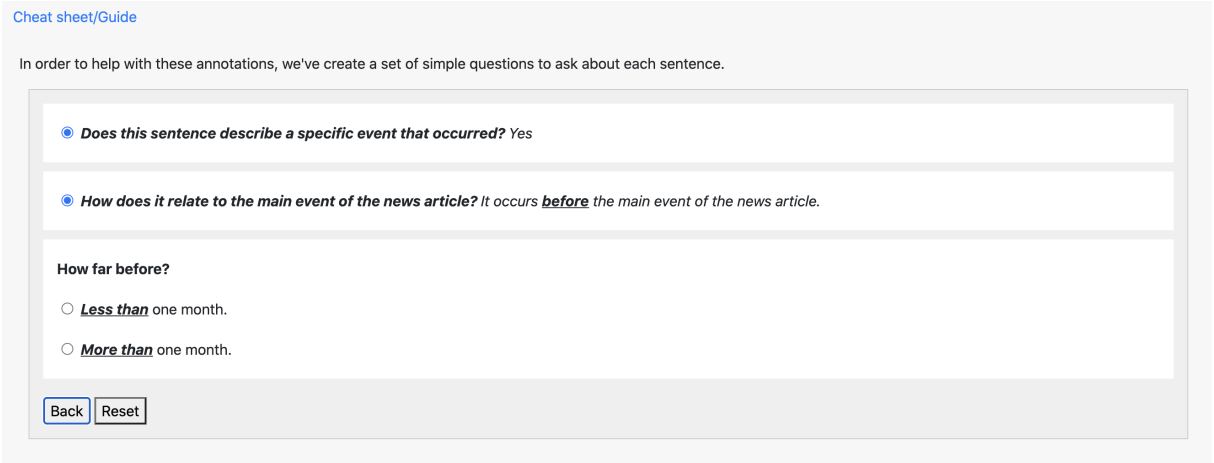


Figure 14: Visual of the training test task that our expert annotators used to solidify their knowledge of the Van Dijk discourse tags (Van Dijk, 2013).

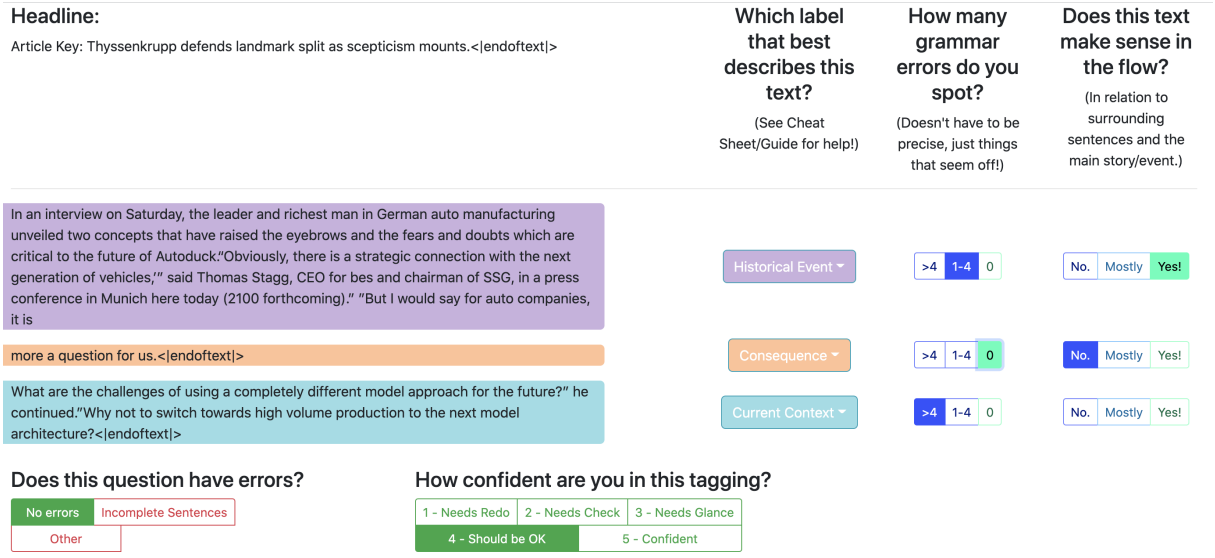


Figure 15: Visual of the Round 1 annotation task interface that we asked our annotators to use to tag. We ultimately DID NOT use this for tagging, and used the task shown in Round 2.

Headline	s_idx	text	class_label	Gold	Grammar-score 1: >6 errors 3: 2-4 errors 5: 0 errors	Global Coherence 1: Poor 3: OK 5: Great	Fits Headline 1: Not at all 3: OK 5: Great
Thyssenkrupp defends landmark split as scepticism mounts	0	a tungus tungi.<|endoftext|>	Main Event	0	1	1	1
Thyssenkrupp defends landmark split as scepticism mounts	1	Tutti uno - tutti, quali unum semissimo.In the case of an ordinarily balanced individual whose body has been balanced in an appropriate manner, his soul is balanced with that balance and so he can enter into communion at any time after death.''<|endoftext|>	Previous Event	0	2	3	1
Thyssenkrupp defends landmark split as scepticism mounts	2	But if he enters the state of sainthood, that soul is not balanced at all.''It's like saying ''the body in itself does nothing.<|endoftext|>	Main Event	0	5	4	1
Thyssenkrupp defends landmark split as scepticism mounts	3	'' It's not balanced at all.''<|endoftext|>	Evaluation	1	4	4	1
Thyssenkrupp defends landmark split as scepticism mounts	4	It's like saying that a person can have both good and evil, but if the body is balanced in an appropriate way it doesn't matter.''<|endoftext|>	Main Event	0	4	4	1
Thyssenkrupp defends landmark split as scepticism mounts	5	In other words, we have to accept the possibility of both good and evil.<|endoftext|>	Main Event	0	4	4	1
Thyssenkrupp defends landmark split as scepticism mounts	6	In other words, if a person is balanced in the correct way, then there will always be good and evil.<|endoftext|>	Main Event	0	4	4	1

Figure 16: Visual of the Round 2 annotation task interface that we asked our annotators to use. This one presented annotators with class labels and asked them to simply determine Y/N whether the label was accurate. We also added a question to probe topicality. We used this interface because assigning labels was ultimately too challenging to make any meaningful distinctions between generation methods. (**Prompting Baseline** is the method generating the text currently seen in the interface.)