MReD: A Meta-Review Dataset for Structure-Controllable Text Generation

Anonymous ACL submission

Abstract

When directly using existing text generation datasets for controllable generation, we are facing the problem of not having the domain knowledge and thus the aspects that could be controlled are limited. A typical example is when using CNN/Daily Mail dataset for controllable text summarization, there is no guided information on the emphasis of summary sentences. A more useful text generator should leverage both the input text and the control signal to guide the generation, which can only be built with deep understanding of the domain knowledge. Motivated by this vision, our paper introduces a new text generation dataset, named MReD. Our new dataset consists of 7,089 meta-reviews and all its 45k meta-review sentences are manually annotated with one of the 9 carefully defined categories, including abstract, strength, decision, etc. We present experimental results on start-of-the-art summarization models, and propose methods for structure-controlled generation with both extractive and abstractive models using our annotated data. By exploring various settings and analyzing the model behavior with respect to the control signal, we demonstrate the challenges of our proposed task and the values of our dataset MReD. Meanwhile, MReD also allows us to have a better understanding of the meta-review domain. [1]

1 Introduction

Text generation entered a new era because of the development of neural network based generation techniques. Along the dimension of the mapping relation between the input information and the output text, we can roughly group the recent tasks into three clusters: more-to-less, less-to-more, and neck-to-neck. The more-to-less text generation tasks output a concise piece of text from some more abundant input, such as text summarization (Tan et al., 2017; Kryściński et al., 2018). The less-to-more generation tasks generate a more abundant output from some obviously simpler input, such as prompt-based story generation (Fan et al., 2018b). The neck-to-neck generation aims at generating an output text which conveys the same quantity of knowledge as the input but in natural language, such as typical RDF triples to text tasks (Gardent et al., 2017).

meta-review:
[This paper studies n-step returns in off-policy RL and introduces a novel algorithm which adapts the return's horizon n in function of a notion of policy's age.]←ABSTRACT [Overall, the reviewers found that the paper presents interesting observations and promising experimental results.]←STRENGTH [However, they also raised concerns in their initial reviews, regarding the clarity of the paper, its theoretical foundations and its positioning (notably regarding the bias/variance tradeoff of uncorrected n-step returns) and parts of the experimental results.]←WEAKNESS [In the absence of rebuttal or revised manuscript from the authors, not much discussion was triggered.]←REBUTTAL PROCESS [Based on the initial reviews, the AC cannot recommend accepting this paper, but the authors are encouraged to pursue this interesting research direction.]←DECISION

Table 1: An example of annotated meta-review. CATEGORY indicates the category of each sentence.

To some extent, the existing task settings are not so adequate because they do not have deep understanding of the domains they are working on, i.e., domain knowledge. Taking text summarization as an example, the most well-experimented dataset CNN/Daily Mail (Nallapati et al., 2016) is composed of the training pairs of news content and news titles. However, it does not tell why a particular piece of news content should have that corresponding title, for example for the same earnings report, why one media emphasizes its new business success in the title, but another emphasizes its net income. Obviously, there is not a standard answer regarding right or wrong. For such cases, if we can specify a control signal, e.g., "emphasizing new business", the generated text would make more sense to users using the text generator.

[1] We will release our code and data at "anonymous URL".

To allow controlling not only the intent of a single generated sentence but also the whole structure of a generated passage, we prepare a new dataset MReD (short for Meta-Review Dataset) with in-depth understanding of the structure of meta-reviews in a peer-reviewing system, namely the open review system of ICLR. MReD for the first time allows a generator to be trained by simultaneously taking the text (i.e. reviews) and the structure control signal as input to generate a meta-review which is not only derivable from the reviews but also complies with the control intent. Thus from the same input text, the trained generator can generate varied outputs according to the given control signal. For example, if the area chair is inclined to accept a borderline paper, he or she may invoke our generator with a structure of "abstract | strength | decision" to generate a meta-review, or may use a structure of "abstract | weakness | suggestion" otherwise. Note that for ease of preparation and explanation, we ground our dataset in the peer review domain. However, the data preparation methodology and proposed models are transferable to other domains, which is indeed what we hope to motivate with this effort.

Specifically, we collect 7,089 meta-reviews of ICLR in recent years (2018 - 2021) and fully annotate the dataset. Each sentence in a meta-review is classified into one of the 9 pre-defined intent categories: abstract, strength, weakness, rating summary, area chair (AC) disagreement, rebuttal process, suggestion, decision, and miscellaneous (misc). Table 1 shows an annotated example, where each sentence is classified into a single category that best describes the intent of this sentence. Our MReD is obviously different from previous text generation/summarization datasets because, given the rich annotations of individual meta-review sentences, a model is allowed to learn more sophisticated generation behaviors to control the structure of the generated passage. Our proposed task is also noticeably different from existing controllable text generation tasks (e.g., text style transfer on sentiment polarity (Shen et al., 2017; Liao et al., 2018) and formality (Shang et al., 2019)) because we focus on controlling the macro structure of the whole passage, rather than the wordings.

To summarize, our contributions are as follows. (1) We introduce a fully-annotated meta-review dataset to make better use of the domain knowledge for text generation. Thorough data analysis provides useful insights into the domain characteristics. (2) We propose a new task of controllable generation focusing on controlling the passage macro structures. It offers stronger generation flexibility and applicability for practical use cases. (3) We design simple yet effective control methods that are independent of the model architecture. We show the effectiveness of enforcing different generation structures with a detailed model analysis. We will release our full dataset, code, and detailed settings to the community.

Year	#Submissions	#withReviews	#Meta-Reviews
2018	994	942	892
2019	1,689	1,639	1,412
2020	2,595	2,517	2,169
2021	2,616	2,616	2,616
Total	7,894	7,714	7,089

Table 2: Dataset statistics of MReD.

2 MReD: Meta-Review Dataset

In this paper, we explore a new task, named the structure-controllable text generation, in a new domain, namely the meta-reviews in the peer reviewing system. Unlike previous datasets that mainly focus on domains like news, meta-review is a worth-studying domain containing essential and high-density opinions. Specifically, during the peer review process of scientific papers, a senior reviewer or area chair will recommend a decision and manually write a meta-review to summarize the opinions from different reviews written by the reviewers. We first introduce the data collection process and then describe the annotation details, followed by dataset analysis.

2.1 Data Collection

We collect the meta-review related data from an online peer reviewing platform for ICLR [2] from 2018 to 2021. Note that the submissions from earlier years are not collected because their meta-reviews are not released. To prepare our dataset for controllable text generation, for each submission, we collect multiple reviews with reviewer ratings and confidence scores, the final meta-review decision, and the meta-review passage. Table 2 shows the statistics of data collected from each year. Initially, 7,894 submissions are collected. After filtering, 7,089 meta-reviews are retained with their corresponding 23,675 reviews. Note that even without

[2] https://openreview.net/

Categories	Definitions
abstract	A piece of summary about the contents of the submission
strength	Reviewers' opinions about the submission's strengths
weakness	Reviewers' opinions about the submission's weaknesses
rating summary	A summary about reviewers' rating scores or decisions
ac disagreement	Area chair (AC) shares different opinions to reviewers
rebuttal process	Contents related to authors' rebuttal with respect to reviews or discussions between reviewers in the rebuttal period
suggestion	Concrete suggestions for improving the submission
decision	Final decision (i.e., accept or reject) on the submission
miscellaneous	None of the above, such as courtesy expressions.

Table 3: Category definition of meta-review sentences.

any further annotation, the dataset can already naturally serve the purpose of multi-document summarization (MDS). Compared with those conventional datasets for MDS, such as TAC (Owczarzak and Dang, 2011) and DUC (Over and Yen, 2004), which contain in total a few hundred input articles (equivalent to reviews in MReD), our dataset is more than 10 times larger.

2.2 Data Annotation

As aforementioned, the structure-controllable text generation aims at controlling the structure of the generated passage. Therefore, we need to comprehensively understand the structures of meta-reviews so as to enable a model to learn how to generate outputs complying with certain structures.

Specifically, based on the nature of meta-reviews, we pre-define 9 intent categories: abstract, strength, weakness, suggestion, rebuttal process, rating summary, area chair (AC) disagreement, decision, and miscellaneous (misc). Table 3 shows the definition for each category (see example sentences in Appendix A.1). The identification of category for some sentences is fairly straightforward, while some sentences are relatively ambiguous. Therefore, besides following the definition of each category, the annotators are also required to follow the additional rules as elaborated in Appendix A.2

For conducting the annotation work, 14 professional data annotators from a data company are initially trained, and 12 of them are selected for the task according to their annotation quality during a trial round. These 12 annotators are fully paid for their work. Each meta-review sentence is independently labeled by 2 different annotators, and a third annotator resolves any disagreement between the first two annotators. We label 45,929 sentences from 7,089 meta-reviews in total, and the Cohen's kappa is 0.778 between the two annotators, showing that the annotation is of quite high quality.

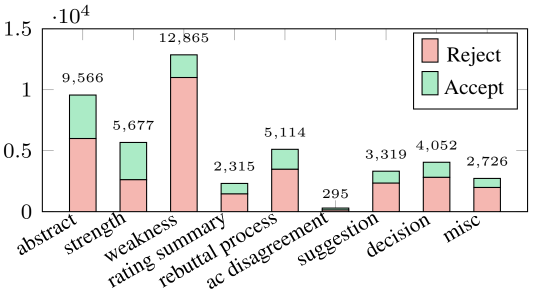
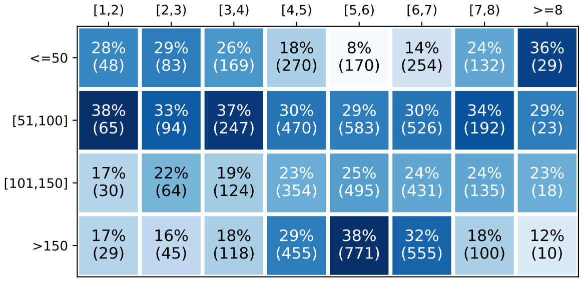


Figure 1: Sentence numbers in different categories.



	[1,2)	[2,3)	[3,4)	[4,5)	[5,6)	[6,7)	[7,8)	>=8
<=50	28% (48)	29% (83)	26% (169)	18% (270)	8% (170)	14% (254)	24% (132)	36% (29)
[51,100]	38% (65)	33% (94)	37% (247)	30% (470)	29% (583)	30% (526)	34% (192)	29% (23)
[101,150]	17% (30)	22% (64)	19% (124)	23% (354)	25% (495)	24% (431)	24% (135)	23% (18)
>150	17% (29)	16% (45)	18% (118)	29% (455)	38% (771)	32% (555)	18% (100)	12% (10)

Figure 2: Meta-review length distribution across ratings. Bracketed numbers show the submission count.

2.3 Data Analysis

To better understand the MReD dataset, we conduct the following analysis along different dimensions.

Sentence distribution across categories. The sentence numbers in different categories are shown in Figure 1, breakdown by the decision (i.e., accept or reject). Among 7,089 submissions, there are 2,368 accepted and 4,721 rejected. Among all submissions and the rejected submissions, "weakness" accounts for the largest proportion, while across the accepted ones, "abstract" and "strength" take up a great proportion. To some extent, these three categories which dominate in meta-reviews could be easily summarized from the reviewers' comments. However, some minor or subjective categories (e.g., "ac disagreement") are hard to generate.

Breakdown analysis by meta-review lengths and average rating scores. We present the percentage of meta-reviews of different lengths in each score range, as shown in Figure 2. For example, among the meta-reviews that receive the reviewers' average score below 2 (i.e., the first column in the figure), 28% are less than or equal to 50 words, and 38% fall in the length range of 51 to 100 words. We can observe that the meta-reviews tend to be longer for those submissions receiving scores in the middle range, while shorter for those with lower scores or higher scores. This coincides with our commonsense that for high-score and low-score sub-

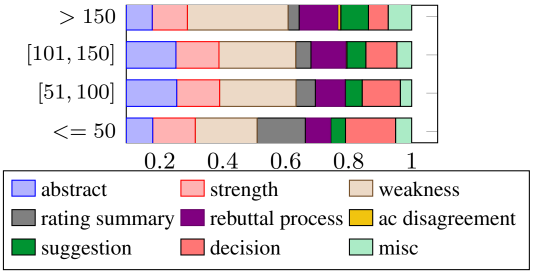


Figure 3: Sentence-level category distribution percentage breakdown by different lengths of meta-reviews.

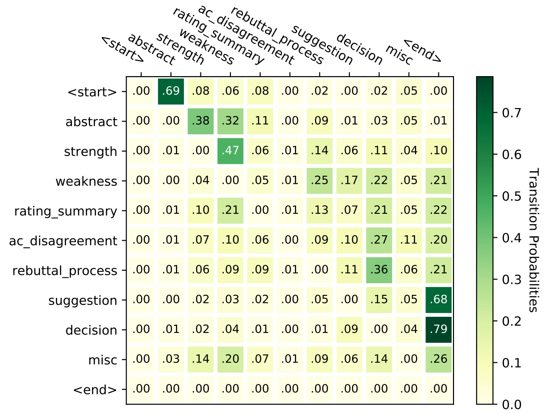
missions, the decision tends to be a clear accept or reject so that meta-reviews can be relatively shorter, while for those borderline submissions, area chairs have to carefully weigh the pros and cons to make the final decision (see Appendix B.1 for borderline submission analysis). As shown in Figure 3, the meta-reviews with more than 150 words generally have a larger proportion of sentences describing "weakness" and "suggestion" for authors to improve the submissions. Additional analysis on the category breakdown for accepted and rejected papers across the score ranges is shown in Appendix B.2.

Meta-review patterns. To study the common structures of meta-reviews, we present the transition matrix of different category segments in Figure 4, where the sum of each row is 1. Note that each segment represents the longest consecutive sentences with the same category. We add "<start>" and "<end>" tokens before and after each meta-review accordingly to investigate which categories tend to be at the start/end of the meta-reviews. It is clear to see that "abstract" usually positions at the beginning of the meta-review, while "suggestion" and "decision" usually appear at the end. There are also some clear patterns appearing in the meta-reviews, such as "abstract | strength | weakness", "rating summary | weakness | rebuttal process", and "abstract | weakness | decision".

3 Structure-Controllable Text Generation

3.1 Task Definition

As aforementioned, in uncontrolled generation, users cannot instruct the model to emphasize on desired aspects. However, in a domain such as meta-reviews, given the same review inputs, one AC may emphasize more on the "strength" of the paper following a structure of "abstract | strength | decision", whereas another AC may prefer a different structure with more focus on reviewers' opinions and suggestions (i.e., "rating summary" and



	<start>	abstract	strength	weakness	rating_summary	ac_disagreement	rebuttal_process	suggestion	decision	misc	<end>
<start>	.00	.69	.08	.06	.08	.00	.02	.00	.02	.05	.00
abstract	.00	.00	.38	.32	.11	.00	.09	.01	.03	.05	.01
strength	.00	.01	.00	.47	.06	.01	.14	.06	.11	.04	.10
weakness	.00	.00	.04	.00	.05	.01	.25	.17	.22	.05	.21
rating_summary	.00	.01	.10	.21	.00	.01	.13	.07	.21	.05	.22
ac_disagreement	.00	.01	.07	.10	.06	.00	.09	.10	.27	.11	.20
rebuttal_process	.00	.01	.06	.09	.09	.01	.00	.11	.36	.06	.21
suggestion	.00	.00	.02	.03	.02	.00	.05	.00	.15	.05	.68
decision	.00	.01	.02	.04	.01	.00	.01	.09	.00	.04	.79
misc	.00	.03	.14	.20	.07	.01	.09	.06	.14	.00	.26
<end>	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

Figure 4: Transition matrix of different categories.

"suggestion"). To achieve such flexibility, the task of structure-controllable text generation is defined as: given the text input (i.e., reviews) and a control sequence of the output structure, a model should generate a meta-review which is derivable from the reviews and presents the required structure.

3.2 Explored Methods

As the recent generation works (Vaswani et al., 2017; Liu and Lapata, 2019; Xing et al., 2020) basically adopt an encoder-decoder based architecture and achieve state-of-the-art performance on many tasks and datasets, we primarily investigate the performance of such a framework on our task. Thus in this subsection, we mainly present how to re-organize the input reviews and the control structure as an input sequence of the encoder. We also explore other baselines in the experiments later.

In order to summarize multiple reviews into a meta-review showing a required structure, we explicitly specify the control label sequence that a model should comply with during generation. Specifically, we intuitively add the control sequence in front of the input text. By directly combining both the control and textual information as a single input, our control method is independent of any specially designed encoder and decoder structures. Moreover, by placing the short control sequence in front, an encoder can immediately observe the control signal at the very beginning, thus avoids the possible interference by the subsequent sequence. Moreover, the control sequence in front will never be truncated when the encoder truncates the input to a certain length limit.

Given the multiple review inputs, we need to linearize them into a single input. One simple method to combine multiple inputs for encoder-decoder models is to concatenate all inputs one after an-

Combination	Obtained Text Input
rate-concat	R1 rating score: S_1, R2 rating score: S_2, R3 rating score: S_3. Review1 <REVBREAK> Review2 <REVBREAK> Review3

Control	Examples of Encoder Input
sent-ctrl	abstract | abstract | decision ==> [TEXT INPUT]
seg-ctrl	abstract | decision ==> [TEXT INPUT]
unctrl	[TEXT INPUT]

Table 4: Upper: example for the review combination method. S_i represents the score given by reviewer Ri. <REVBREAK> is the special separator used to concatenate different review texts. Lower: examples of control methods. [TEXT INPUT] refers to the obtained text from the upper section.

other (Fabbri et al., 2019). Beside the text inputs, the review rating is also crucial information for writing meta reviews, which cannot be found in the review passages but exists in the field of rating score. Therefore, we create a rating sentence that consists of the extracted ratings given by the corresponding reviewers and prepend it to our concatenated review texts to obtain the final input. We name this method ***rate-concat*** (see Table 4, upper). We also show explorations with other review combination methods in Appendix C.1.

As aforementioned, we place the control sequence in front of the re-organized review information. Specifically, we explore two different control methods, namely, ***sent-ctrl*** and ***seg-ctrl***. *Sent-ctrl* uses one control label per target sentence and controls generation on a sentence-level. Note that this method can allow implicit control on the length (i.e., number of sentences) of the generation. *Seg-ctrl* treats consecutive sentences of the same label as one segment and only uses one label for a single segment. Example inputs of different control settings are shown in Table 4 (lower). For instance, sent-ctrl repeats "abstract" in its control sequence whereas seg-ctrl does not. This is because seg-ctrl treats the 1st and 2nd target sentences of "abstract" as the same segment and only uses a single label to indicate it in the sequence. Additionally, we provide a vanilla setting for uncontrolled generation, ***unctrl***, where no control sequence is used.

Using the above input sequence as the source and the corresponding meta-review as the target, we can train an encoder-decoder model for controllable generation. Many transformer-based models have achieved state-of-the-art performance. Common abstractive summarization models include BART (Lewis et al., 2020), T5 (Raffel et al., 2020) and PEGASUS (Zhang et al., 2020). In this paper we focus on the *bart-large-cnn* model, one variant of the BART model (results on other pretrained models can be found in Appendix D.1). More specifically, we use the pytorch implementation in the open-source library Hugging Face Transformers (Wolf et al., 2020). Hence, all our future usage of the word "Transformers" refers to *bart-large-cnn* in the Transformers library .

4 Experiments

4.1 Baselines

Extractive Baselines. We employ three common extractive summarization baselines each of which basically provides a mechanism to rank the input sentences. ***LexRank*** (Erkan and Radev, 2004) represents sentences in a graph and uses eigenvector centrality to calculate sentence importance scores. ***TextRank*** (Mihalcea and Tarau, 2004) is another graph-based sentence ranking method that obtains vertex scores by running a "random-surfer model" until convergence. ***MMR*** (Carbonell and Goldstein, 1998) calculates sentence scores by balancing the redundancy score with the information relevance score. After ranking with the above models, we select sentences as output with different strategies according to the controlled and uncontrolled settings. For the uncontrolled setting, we simply select the top k sentences as the generated output, where k is a hyperparameter deciding the size of the generated output. For the controlled setting, we select only the top sentences with the right category labels according to the control sequence. To do so, we employ an LSTM-CRF (Lample et al., 2016) tagger trained on the labeled meta-reviews to predict the sentence labels of each input review. Refer to Appendix D.2 for more details of the tagger.

Generic Sentence Baselines. Considering the nature of meta-reviews, we could imagine some categories may have common phrases inflating the Rouge scores, such as "This paper proposes ..." for abstract, and "I recommend acceptance." for decision, etc. To examine such impact, we select sentences that are generic in each category and combine these sentences to generate outputs according to the control sequences. For instance, if the control sequence is "abstract | strength | decision", we take the most generic sentences from the categories of "abstract", "strength" and "decision" respectively to form the output. Specifically, we create two generic sentence baselines by obtaining

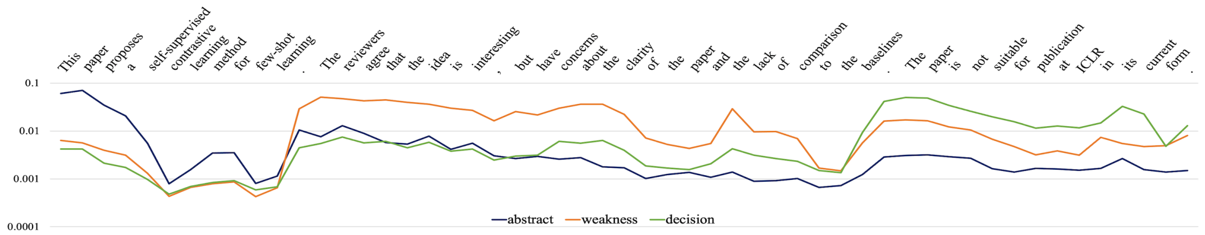


Figure 5: Cross attention weights of each generated token towards the control tokens in logarithmic scale.

	R_1	R_2	R_L
Source Generic	23.00	3.19	11.56
Target Generic	**31.24**	**5.92**	**16.49**
MMR, *unctrl*	30.98	5.36	16.07
LexRank, *unctrl*	31.42	6.62	16.39
TextRank, *unctrl*	32.22	7.24	16.81
MMR, *sent-ctrl*	32.33	6.39	**17.60**
LexRank, *sent-ctrl*	31.94	6.52	16.99
TextRank, *sent-ctrl*	**33.02**	**7.15**	17.49
Transformers, *unctrl*	34.70	8.71	20.92
Transformers, *sent-ctrl*	**38.82**	**10.71**	**23.11**
Transformers, *seg-ctrl*	36.49	9.88	22.76

Table 5: Meta-review generation results on MReD.

generic sentences from the training data from either the meta-review references (i.e., target) or the input reviews (i.e., source), namely "Target Generic" and "Source Generic". Moreover, we also study such impact on the high-score and low-score submissions respectively, since an AC may write more succinct meta-reviews for clear-cut papers, as suggested by Figure 2. See Appendix D.3 for more details and results on generic sentence baselines.

4.2 Experimental Setting

To conduct text generation experiments, we preprocess our MReD dataset by filtering to ensure the selected meta-reviews have 20 to 400 words, as certain meta-review passages are extremely short or long. After preprocessing, we obtain 6,693 source-target pairs, for which we randomly split into train, validation, and test sets by a ratio of 8:1:1. We evaluate our generated outputs against the reference meta-reviews using the F_1 scores of $ROUGE_1$, $ROUGE_2$, and $ROUGE_L$ (Lin, 2004) [3]. For the extractive and generic baselines, a key hyperparameter is the sentence number k, which we set to the number of labels in the *sent-ctrl* control sequence. More setting details are shown in Appendix D.4

4.3 Main Results

We show results in Table 5. Only the best settings of *rate-concat* (Table 12 in Appendix C.1) and input truncation of 2048 tokens (Appendix D.5) for the Transformers are included. Amongst the extractive baselines, TextRank performs the best in both *unctrl* and *sent-ctrl* settings. Nevertheless, all controlled methods outperform their *unctrl* settings (same for the Transformers). This validates our intuition that structure-controlled generation is more suitable for user-subjective writings such as meta-reviews, because the model can better satisfy different structure requirements when supplied with the corresponding control sequences. On the other hand, for the Transformers, *sent-ctrl* is the best, followed by *seg-ctrl*. This is most likely due to the former's more fine-grained sentence-level control that provides a clearer structure outline, as compared to the coarser segment-level control.

Moreover, the Transformers far outperform the extractive baselines, showing that the extraction-based methods are insufficient for MReD. This also suggests that meta-review writings are different from the input reviews, therefore copying full review sentences to form meta-reviews doesn't work well. This is again validated by the "Target Generic" baseline's significant improvement over the "Source Generic" baseline, which shows that generic sentences from meta-reviews can suit generation much better than those in reviews. Nevertheless, all Transformers results are still much better than the "Target Generic" sentence baseline, showing that despite generic phrases in some categories contributing to Rouge, the Transformers model is capable of capturing content-specific information for each input.

4.4 Case Study

We study some cases for a better understanding of the structure-controllable generation.

Identify the control label for each sentence. We first evaluate whether the model is able to attend to the correct control label during generation. For each generation step, we obtain the cross attention weights from the decoder's output token towards the control labels and plot them in Figure 5. The given control sequence is "abstract | weakness |

[3] We use the Hugging Face Transformers' Rouge evaluation script, which has the field "use_stemmer" enabled.

	Gold Labels	Gold Meta-review
0	abstract ǀ weakness ǀ decision	[The paper presents a self-supervised model based on a contrastive autoencoder that can make use of a small training set for upstream multi-label/class tasks.]←ABSTRACT [Reviewers have several concerns, including the lack of comparisons and justification for the setting, as well as the potentially narrow setting.]←WEAKNESS [Overall, I found the paper to be borderline, the cons slightly greater than the pros, so I recommend to reject it.]←DECISION
	Control Sequence	**Generated Passage**
1	abstract ǀ weakness ǀ decision	[This paper proposes a self-supervised contrastive learning method for few-shot learning.]←ABSTRACT [The reviewers agree that the idea is interesting, but have concerns about the clarity of the paper and the lack of comparison to the baselines.]←WEAKNESS [The paper is not suitable for publication at ICLR in its current form.]←DECISION
2	abstract ǀ abstract ǀ weakness ǀ decision	[This paper proposes a self-supervised contrastive learning method for few-shot learning.]←ABSTRACT [The idea is to learn the label embeddings using in-task data, and then use those for training a contrastive autoencoder for multi-label classification on the long-tail problem.]←ABSTRACT [The reviewers and AC note the following potential weaknesses: (1) limited novelty, (2) limited evaluation, and (3) lack of comparison to relevant baselines.]←WEAKNESS [AC agrees with the reviewers that the paper is not ready for publication in its current form.]←DECISION
3	strength ǀ weakness ǀ weakness ǀ rebuttal process ǀ suggestion ǀ decision	[Pros: The paper is well-motivated and the idea is simple and straightforward.]←STRENGTH [However, the reviewers have raised several concerns about the evaluation and the clarity of the paper.]←WEAKNESS [In particular, the lack of comparison to the baselines is a major concern.]←WEAKNESS [The authors have addressed some of these concerns in the rebuttal, but the reviewers are still not convinced about the significance of the results.]←REBUTTAL PROCESS [The paper would be much stronger if the authors could compare their method to more baselines for zero-shot learning, such as matching networks and Siamese networks.]←SUGGESTION [Overall, the paper is not ready for publication at ICLR.]←DECISION

Table 6: Varied generation outputs by giving different control sequences.

Generated Content	Attention Attribution
Sent 1 (abstract): This paper proposes a self-supervised contrastive learning method for few-shot learning.	Control: abstract ǀ weakness ǀ decision ==> Top 1: This paper proposes a contrastive autoencoder approach that only requires small data to perform a multi-label classification on the long-tail problem. Top 2: The goal of this work is to enable zero-shot and few-shot learning with very few resources as a more sustainable approach to machine learning applications. <sep> Top 3: I like the intended focus of this paper which is to perform self-supervised training of small data for downstream tasks with applications for zero and few-shot learning.
Sent 2 (weakness): The reviewers agree that the idea is interesting, but have concerns about the clarity of the paper and the lack of comparison to the baselines.	Control: abstract ǀ weakness ǀ decision ==> Top 1: The key concern about the paper is the lack of comparison to the baselines. Top 2: However, I still have issues with the evaluation and the clarity of the paper. Top 3: In general, the idea is interesting, since it leverages the in-task data for distinguishing positive and negative samples.
Sent 3 (decision): The paper is not suitable for publication at ICLR in its current form.	Control: abstract ǀ weakness ǀ decision ==> Top 1: I think the paper requires another round of revision before it is ready for publication. Top 2: The proposed idea is very straightforward by combining a matching network with contrastive learning to give broader signals. Top 3: In general, the idea is interesting, since it leverages the in-task data for distinguishing positive and negative samples.

Table 7: Attention analysis for each output sentence.

decision”. When generating each sentence, we can see that the attention weights of the corresponding control token are the highest, which demonstrates that our model can effectively pay attention to the correct control label and thus generate the content complying with the intent.

Extract information from the input sentences. To understand what information the model attends to when generating each sentence, we aggregate the cross attention weights to obtain the attention scores from each generated sentence towards all input sentences (Appendix D.6). Then, we select the top 3 input sentences with the highest attention scores for each generated sentence, and visualize the normalized attention weights on all tokens in the selected sentences and the control sequence in Table 7. As shown, the model can correctly extract relevant information from the source sentences. For example, it identifies important phrases such as “interesting”, “clarity” and “lack of comparison to baselines” when generating “Sent 2”.

Generate varied outputs given different control sequences. To further investigate the effectiveness of the control sequence, we change the control sequence of the above example and re-generate the meta-reviews given the same input reviews. In Table 6, we first show the gold meta-review and the model output using the original control sequence in Row 0 and Row 1, and then show the model outputs with alternative control sequences in Row 2 and Row 3. From the outputs, we can see that indeed each generated sentence corresponds to its control label well. In Row 2, we add an additional control label in the sequence and by repeating the “abstract” label, the generator can further elaborate more details of the studied method. This is one key advantage of our *sent-ctrl* compared to the *seg-ctrl*, which allows the control of length and the level of the generation details. In Row 3, a very comprehensive control sequence is specified. We can see that the output meta-review is quite fluent and polite to reject the borderline paper. See Appendix D.7 for more examples.

4.5 Human Evaluation

In addition to the Rouge evaluation, we ask 3 human judges to manually assess the generation quality of the Transformers models from Table 5 on 100 random test instances. For each test instance, we provide the judges with the input reviews and randomly ordered generations from different models,

	Unctrl	Sent-ctrl	Seg-ctrl
Fluency	4.145	**4.630***	4.090
Content Relevance	**4.585**	4.335	4.410
Structure Similarity (sent)	0.298	**0.706***	-
Structure Similarity (seg)	0.363	-	**0.623***
Decision Correctness	0.685	**0.830***	0.695

Table 8: Human evaluation. * indicates the ratings of corresponding models significantly (by Welch's t-test) outperform the *unctrl*: $p < 0.01$ for decision correctness, $p < 0.0001$ for fluency and structure similarity.

and ask them to individually evaluate the generations based on the following criteria: (1) *Fluency*: is the generation fluent, grammatical, and without unnecessary repetitions? (2) *Content Relevance*: does the generation reflect the review content well, or does it produce general but trivial sentences? (3) *Structure Similarity*: how close does the generation structure resemble the gold structure (i.e., the control sequence)? (4) *Decision Correctness*: does the generation agree with the gold human decision? We grade fluency and content relevance on a scale of 1 to 5, whereas structure similarity and decision correctness are calculated from 0 to 1 (Appendix D.8). For structure similarity, because *sent-ctrl* and *seg-ctrl* have different control sequences, we evaluate the two models on sentence-level (sent) and segment-level (seg) structures respectively, and provide both evaluations for *unctrl*.

As shown in Table 8, both *sent-ctrl* and *seg-ctrl* models show significant improvements on the generation structure over the uncontrolled baseline, which affirms the effectiveness of our proposed methods for structure-controllable generation. *Sent-ctrl* also has better fluency and decision correctness, suggesting that having a better output structure can benefit the readability and decision generation. For the content relevance, the scores of all methods are reasonably good, and significance tests cannot prove any best model ($p > 0.08$). Nevertheless, it is possible that the looser control a method applies, the better relevance score it achieves. It is because a tighter control narrows the content that a model can use from the reviews.

5 Related Work

To facilitate the study of text summarization, earlier datasets are mostly in the news domain with relatively short input passages, such as NYT (Sandhaus, 2008), Gigaword (Napoles et al., 2012), CNN/Daily Mail (Hermann et al., 2015), NEWSROOM (Grusky et al., 2018) and XSUM (Narayan et al., 2018). Datasets for long documents include Sharma et al. (2019), Cohan et al. (2018), and Fisas et al. (2016). In this paper, we explore text summarization in a new domain (i.e., the peer review domain) and provide a new dataset, i.e., MReD. Moreover, MReD's reference summaries (i.e., meta-reviews) are fully annotated and thus allow us to propose a new task, namely structure-controllable text generation.

Researchers recently explore the peer review domain data for a few tasks, such as PeerRead (Kang et al., 2018) for paper decision predictions, AMPERE (Hua et al., 2019) for proposition classification in reviews, and RR (Cheng et al., 2020) for paired-argument extraction from review-rebuttal pairs. Additionally, a meta-review dataset is introduced by Bhatia et al. (2020) without any annotation. There are also some explorations on research articles (Teufel et al., 1999; Liakata et al., 2010; Lauscher et al., 2018), which differ in nature from the peer review domain.

A wide range of control perspectives has been explored in controllable generation, including style control (e.g., sentiments (Duan et al., 2020), politeness (Madaan et al., 2020), formality (Wang et al., 2019), domains (Takeno et al., 2017) and persona (Zhang et al., 2018)) and content control (e.g., length (Duan et al., 2020), entities (Fan et al., 2018a), and keywords (Tang et al., 2019)). Our structure-controlled generation differs from these works as we control the high-level output structure, rather than the specific styles or the surface details of which keywords to include in the generated output. Our task also differs from content planning (Reiter and Dale, 1997; Shao et al., 2019; Hua and Wang, 2019), which involves explicitly selecting and arranging the input content. Instead, we provide the model with the high-level control labels, and let the model decide on its own the relevant styles and contents.

6 Conclusions

This paper introduces a fully-annotated text generation dataset MReD in a new domain, i.e., the meta-reviews in the peer review system, and provides thorough data analysis to better understand the data characteristics. With such rich annotations, we propose simple yet effective methods for structure-controllable text generation. Extensive experimental results are presented as baselines for future study and thorough result analysis is conducted to shed light on the control mechanisms.

References

Chaitanya Bhatia, Tribikram Pradhan, and Sukomal Pal. 2020. Metagen: An academic meta-review generation system. In *Proceedings of ACM-SIGIR*.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of ACM-SIGIR*.

Liying Cheng, Lidong Bing, Qian Yu, Wei Lu, and Luo Si. 2020. Argument pair extraction from peer review and rebuttal via multi-task learning. In *Proceedings of EMNLP*.

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 NAACL*.

Yuguang Duan, Canwen Xu, Jiaxin Pei, Jialong Han, and Chenliang Li. 2020. Pre-train and plug-in: Flexible conditional text generation with variational autoencoders. In *Proceedings of the ACL*.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Artificial Intelligence Research*.

Alexander Richard Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of ACL*.

Angela Fan, David Grangier, and Michael Auli. 2018a. Controllable abstractive summarization. In *Proceedings of WNGT*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018b. Hierarchical neural story generation. In *Proceedings of ACL*.

Beatriz Fisas, Francesco Ronzano, and Horacio Saggion. 2016. A multi-layered annotated corpus of scientific papers. In *Proceedings of LREC*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from rdf data. In *Proceedings of INLG*.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of NAACL*.

Karl Moritz Hermann, Tomás Kociskỳ, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of NIPS*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*.

Xinyu Hua, Mitko Nikolov, Nikhil Badugu, and Lu Wang. 2019. Argument mining for understanding peer reviews. In *Proceedings of NAACL*.

Xinyu Hua and Lu Wang. 2019. Sentence-level content planning and style specification for neural text generation. In *Proceedings of EMNLP-IJCNLP*.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and nlp applications. In *Proceedings of NAACL*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv e-prints*.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of EMNLP*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL*.

Anne Lauscher, Goran Glavaš, and Kai Eckert. 2018. Arguminsci: A tool for analyzing argumentation and rhetorical aspects in scientific writing. In *Proceedings of ACL*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*.

Maria Liakata, Simone Teufel, Advaith Siddharthan, and Colin Batchelor. 2010. Corpora for the conceptualisation and zoning of scientific papers. In *Proceedings of LREC*.

Yi Liao, Lidong Bing, Piji Li, Shuming Shi, Wai Lam, and Tong Zhang. 2018. QuaSE: Sequence editing under quantifiable guidance. In *Proceedings of EMNLP*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of ACL*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv e-prints*.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabás Póczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. 2020. Politeness transfer: A tag and generate approach. In *Proceedings of ACL*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of EMNLP*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of SIGNLL*.

Courtney Napoles, Matthew R Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of AKBC-WEKEX*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of EMNLP*.

Paul Over and James Yen. 2004. An introduction to duc-2004. In *Proceedings of DUC*.

Karolina Owczarzak and Hoa Trang Dang. 2011. Overview of the tac 2011 summarization track: Guided task and aesop task. In *Proceedings of TAC*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

LA Ramshaw. 1995. Text chunking using transformation-based learning. In *Proceedings of Third Workshop on Very Large Corpora*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of CoNLL*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP*.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*.

Mingyue Shang, Piji Li, Zhenxin Fu, Lidong Bing, Dongyan Zhao, Shuming Shi, and Rui Yan. 2019. Semi-supervised text style transfer: Cross projection in latent space. In *Proceedings of EMNLP-IJCNLP*.

Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 2019. Long and diverse text generation with planning-based hierarchical variational model. In *Proceedings of EMNLP-IJCNLP*.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of ACL*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Proceedings of NIPS*.

Shunsuke Takeno, Masaaki Nagata, and Kazuhide Yamamoto. 2017. Controlling target features in neural machine translation via prefix constraints. In *Proceedings of WAT*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of ACL*.

Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric Xing, and Zhiting Hu. 2019. Target-guided open-domain conversation. In *Proceedings of ACL*.

Simone Teufel, Jean Carletta, and Marc Moens. 1999. An annotation scheme for discourse-level argumentation in research articles. In *Proceedings of EACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2019. Harnessing pre-trained neural networks with rules for formality style transfer. In *Proceedings of EMNLP-IJCNLP*.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP*.

Xinyu Xing, Xiaosheng Fan, and Xiaojun Wan. 2020. Automatic generation of citation texts in scholarly papers: A pilot study. In *Proceedings of ACL*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of ICML*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of ACL*.

Categories	Examples
abstract	"The paper presents/explores/describes/addresses/proposes ..."
strength	"The reviewers found the paper interesting." "The method and justification are clear." "The quantitative results are promising."
weakness	"The paper is somewhat incremental ..." "... claims are confusing" "The main concern is ..." "... unfair experimental comparisons ..."
rating summary	"R1 recommends Accept." "All four reviewers ultimately recommended acceptance." "Reviews were somewhat mixed, but also with mixed confidence scores."
ac disagreement	"The area chair considers the remaining concerns by Reviewer 3 as invalid." "I do not agree with the criticism about ..." "I disagree with the second point ..."
rebuttal process	"The authors have made various improvements to the paper" "... remained after the author rebuttal ..." "Authors provided convincing feedbacks on this key point."
suggestion	"... more analysis ..." "The authors are advised to take into account the issues about ..."
decision	"The paper is recommended as a poster presentation." "AC recommends Reject." "I recommend rejection."
miscellaneous	"Thank you for submitting you paper to ICLR." "I've summarized the pros and cons of the reviews below."

Table 9: Category examples of meta-review sentences.

A Data Annotation

A.1 Category definitions

We show category examples in Table 9.

A.2 Additional annotation rules

The additional rules for annotation are as follows: First, instead of only labeling the individual sentences per se, the annotators are given a complete paragraph of meta-review to label the sentences with context information. For example, if the area chair writes a sentence providing some extra background knowledge in the discussion of the weakness of the submission, that sentence itself can be considered as "misc". However, it should be labeled as "weakness" to be consistent in context.

Second, not every sentence can be strictly classified into a single category. When a sentence contains information from multiple categories, the annotators should consider its main point and primary purpose. One example is: "Although the paper discusses an interesting topic and contains potentially interesting idea, its novelty is limited." Although the first half of the sentence discusses the strength of the submission, the primary purpose of this sentence is to point out its weakness, and therefore it should be labeled as weakness.

Furthermore, there are still some cases where the main point of the sentence is hard to differentiate from multiple categories. We then define a priority order of these 9 categories according to the importance of each category for annotators to follow: decision > rating summary > strength $\overset{?}{=}$ weakness > ac disagreement > rebuttal process > abstract > suggestion > miscellaneous. We use the sign "$\overset{?}{=}$" because there are some rare cases where a sentence contains both "strength" and "weakness" while there is no obvious emphasis on either, and it is hard to tell whether "strength" should have a priority over "weakness" or the other way round. We then label this sentence based on the final decision: if this submission is accepted, we label the sentence as "strength", and vice versa.

	Accept	Reject
abstract	23.8%	18.1%
strength	18.1%	9.3%
weakness	13.5%	34.3%
rating summary	6.3%	4.1%
ac disagreement	2.2%	0.5%
rebuttal process	13.2%	11.0%
suggestion	7.7%	8.2%
decision	9.2%	8.1%
miscellaneous	6.2%	6.4%

Table 10: Category distribution of borderline submissions (average score in the range of [4.5,6) breakdown by final decision.

B Data Analysis

B.1 Borderline papers

We further analyze the category distribution in borderline papers. As shown in Table 10, for submissions within the score range of [4.5,6), there are 713 accepted submissions and 2,588 rejected submissions. One clear difference is the percentage of "strength" and "weakness". Another difference is the percentage of "ac disagreement", where the accepted papers have four times the value than rejected ones. This suggests that for the accepted borderline papers, the area chair tends to share different opinions with reviewers, and thus deciding to accept the borderline submissions.

B.2 Percentage of each category for accepted and rejected papers across score ranges

We further analyze the occurrence of each category for accepted papers and rejected papers separately across different score ranges, as shown in Table 11. For accepted papers, as the score increases, the percentage of meta-reviews having "weakness" and "suggestion" drops because the high-score submissions are more likely to be accepted. Even the percentage of "decision" drops following the same trend. In addition, the proportion of meta-reviews

	Accept			Reject		
	Low	Med	High	Low	Med	High
abstract	**79**	75	74	69	69	74
strength	64	**71**	70	26	43	50
weakness	49	44	32	79	84	**88**
rating summary	25	**33**	32	29	25	24
ac disagreement	1	**6**	2	1	2	3
rebuttal process	**52**	47	37	35	39	39
suggestion	29	26	23	23	32	**38**
decision	**56**	53	46	53	53	**56**
miscellaneous	19	19	14	24	35	**45**

Table 11: Occurrence of different categories for accepted and rejected papers, breakdown by average scores. Low for scores ≤ 5.5, high for scores ≥ 6.5, and med for borderline scores in between.

	R_1	R_2	R_L
longest-review	33.00	7.98	20.37
concat	34.12	8.49	20.59
merge	34.42	**8.77**	20.73
rate-concat	**34.70**	8.71	**20.92**
rate-merge	34.40	8.72	20.74

Table 12: Meta-review uncontrolled generation results for different review combination methods.

Pretrained Model	R_1	R_2	R_L
Uncontrolled Generation			
facebook/bart-large-cnn*	**34.70**	8.71	20.92
facebook/bart-large	34.50	8.91	21.10
t5-large	33.51	**9.10**	**22.12**
google/pegasus-cnn_dailymail	31.31	7.03	19.03
Controlled Generation, *sent-ctrl*			
bart-large-cnn*	**38.82**	10.71	23.11
facebook/bart-large	37.04	10.22	23.26
t5-large	36.37	**10.78**	**24.83**
google/pegasus-cnn_dailymail	34.66	9.10	22.21

Table 13: Results of other common Transformers summarization models using source truncation of 2048. * represents our selected model in the main paper.

having “rebuttal process” is larger for submissions with lower scores. This suggests that the rebuttal process plays an important role in the peer review process, especially in helping the borderline papers to be accepted.

On the other hand, for rejected papers, the percentage of meta-reviews having “strength” increases as the average score increases. This coincides with our common sense that the submissions receiving higher scores tend to have more strengths. One interesting finding here is that the percentage of “weakness” and “suggestion” also increases as the average rating score increases. This may be due to two main reasons. First, to reject a submission with higher scores, the area chair has to explain the weakness with more details and provide more suggestions for authors to further improve their submissions. Second, compared to the percentage of “strength”, “weakness” definitely has a larger percentage within any range of rating scores. The difference in the percentage of “strength” and “weakness” is intuitively different between the accepted papers and the rejected papers.

C Structure-Controllable Text Generation

C.1 Review combination methods

We explore alternative methods to linearize the multiple reviews of the same submission, namely, ***concat*** and ***merge***. For the *concat*, we simply concatenate all reviews one after another according to their reviewers’ sequence. For *merge*, we can obtain the merged content as follows: From all review inputs, we use the longest one as a backbone. We segment all reviews’ content on a paragraph level, and encode them using SentenceTransformers (Reimers and Gurevych, 2019). Then, for each paragraph embedding in the non-backbone reviews, we calculate a cosine similarity score with each backbone paragraph embedding, and insert it after the backbone paragraph with which it has the highest similarity score. We repeat the process for all paragraphs in non-backbone reviews to obtain a single passage. Additionally, we provide a baseline setting ***longest-review***, which does not combine reviews but only uses the longest review as the input. Moreover, we add rating sentences in front of the results of *concat* and *merge* to obtain ***rate-concat*** and ***rate-merge***, respectively.

As shown in Table 12, the *longest-review* setting has the worst performance, thus validating that the review combination methods are necessary in order not to omit important information. *rate-concat* setting has the best overall performance, which is the setting used throughout the main paper.

D Experiments

D.1 Additional transformers models

We provide baselines of uncontrolled generation and controlled generation on MReD using other common Transformer pretrained models in Table 13.

D.2 Tagger for source sentences

To obtain labels on source input, we train a tagger based on the human-annotated meta-reviews,

	Micro F$_1$	Macro F$_1$	abstract	strength	weakness	rating	ACdisagree	rebuttal	suggestion	decision	misc
BERT-base-cased + CRF	85.27	76.71	**94.58**	86.12	86.21	85.21	30.77	73.80	73.89	91.30	68.49
BERT-large-cased + CRF	84.68	77.84	93.93	**86.71**	84.36	84.07	40.00	72.60	74.35	91.60	**72.96**
RoBERTa-base + CRF	**85.83**	**79.98**	94.47	86.43	86.73	84.56	**54.84**	**74.44**	72.79	**93.08**	72.54
RoBERTa-large + CRF	85.72	79.34	94.42	85.61	**87.09**	**85.40**	50.00	73.97	**75.63**	90.93	71.00

Table 14: Main results for meta-review discourse understanding.

then use it to predict labels on the input sentences. Specifically, we define the task as a sequence labeling problem and apply the long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) networks with a conditional random field (CRF) (Lafferty et al., 2001) (i.e., LSTM-CRF (Lample et al., 2016)) model on the annotated MReD dataset. The same data split as the meta-review generation task is used. We adopt the standard IOBES tagging scheme (Ramshaw, 1995; Ratinov and Roth, 2009), and fine-tune BERT (Kenton and Toutanova, 2019) and RoBERTa (Liu et al., 2019) models in Hugging Face. All models are trained for 30 epochs with an early stop of 20, and each epoch takes about 30 minutes. We select the best model parameters based on the best micro F$_1$ score on the development set and apply it to the test set for evaluation. All models are run with V100 GPU. We use Adam (Kingma and Ba, 2014) with an initial learning rate of 2e-5.

We report the F$_1$ scores for each category as well as the overall micro F$_1$ and macro F$_1$ scores in Table 14. Micro F1 is the overall accuracy regardless of the categories, whereas macro F1 is an average of per category accuracy evaluation. Since some of the category labels (eg. "ac disagreement") are very rare, their classification accuracy is low. Overall, micro F1 is a more important metric since it suggests general performance. The results stand proof that the majority of the categories have their own characteristics that can be identified from other categories. RoBERTabase is the best performing model, therefore we use this model for review sentence label prediction.

D.3 Generic sentence baselines

Besides the baselines of "Source Generic" and "Target Generic", we explore subsets of papers with high scores (average reviewers' rating $\geqslant 7$) or low scores (average reviewers' rating $\leqslant 3$) to obtain 4 additional generic baselines: "Source High Score", "Source Low Score", "Target High Score", "Target Low Score". We use "Target Generic" as an example to explain how we obtain the generic sentences: We first group all meta-review sentences from the training set according to their label categories, and then re-arrange the sentences in each category using TextRank (our best performing extractive model). Since TextRank ranks the input sentences based on each sentence's content connection with others, sentences with higher rankings are also more general in the sense that they have more shared content with others. Similarly, different sets of generic sentences can be obtained for the other 5 baselines.

	R$_1$	R$_2$	R$_L$
Source Generic	23.00	3.19	11.56
Source High Score	23.37	**3.58**	**12.82**
Source Low Score	**25.28**	2.96	12.77
Target Generic	31.24	5.92	16.49
Target High Score	30.82	5.38	16.21
Target Low Score	**31.70**	**7.30**	**18.55**

Table 15: Meta-review generation results on MReD dataset under Rouge$_1$, Rouge$_2$, and Rouge$_L$ F$_1$ scores for the generic sentence baselines.

After obtaining the generic sentence sets, we can create baseline generations using the *sent-ctrl* sequence. We avoid using the same sentence twice inside the same generation, so if the same label appears multiple times in a control sequence, we will use the same number of generic sentences for that category down the ranking order.

We show results in Table 15. The low score baselines perform the best amongst both source and target baselines, suggesting that the sentences from low score submissions are more typical for both reviews and meta-reviews.

D.4 Experimental setting details

For preprocessing, besides filtering based on meta-review length, we also remove submissions with only one or two reviews, since the majority of the submissions have more than 3 reviews.

For the extractive baselines, recall that under the sent-ctrl setting, the control sequence length is the same as the sentence number of the target meta-review. Therefore, to conduct a fair comparison, we set the hyperparameter k equal to the number of

	Gold Labels	Gold
0	abstract | rating summary	[this work adapts cycle GAN to the problem of decipherment with some success.]←ABSTRACT [it's still an early result, but all the reviewers have found it to be interesting and worthwhile for publication.]←RATING SUMMARY
	Control Sequence	**Generated Passage**
1	abstract | rating summary	[This paper proposes a method for using cycle GANs to decipher text encrypted with historical ciphers.]←ABSTRACT [The reviewers were unanimous in their vote to accept.]←RATING SUMMARY
2	abstract | abstract | rating summary	[This paper proposes a method for using cycle GANs to decipher text encrypted with historical ciphers.]←ABSTRACT [The proposed method is based on replacing the log loss of the GAN with a quadratic loss and regularizing the Jacobian of the discriminator.]←ABSTRACT [The reviewers were unanimous in their vote to accept.]←RATING SUMMARY
3	strength | decision	[The reviewers agree that the paper is well-written, the problem is interesting, and the proposed method is novel.]←STRENGTH [The paper should be accepted.]←DECISION
4	misc | abstract | strength | decision	[Thanks for your submission to ICLR.]←MISC [This paper proposes a method for using cycle GANs to decipher text encrypted with historical ciphers.]←ABSTRACT [This is an interesting and fairly original problem, and the proposed method is shown to be very effective.]←STRENGTH [All reviewers agree that the paper is well written, and I'm happy to recommend acceptance.]←DECISION
5	weakness | rebuttal process	[The reviewers raised a number of concerns including the correctness of the proof, the lack of a simple baseline, and the presentation of the paper.]←WEAKNESS [The authors' rebuttal addressed some of these concerns, but not to the degree that the reviewers felt it should be.]←REBUTTAL PROCESS

Table 16: Generation examples of alternative control sequences on the same review inputs using the *sent-ctrl* method.

Data Split	max	med	avg
train	7276	1482	1368
validation	3762	1427	1352
test	5144	1454	1352

Table 17: Source length statistics on all data splits. Max for maximum source length, med for median source length, and avg for average source length.

length	R_1	R_2	R_L
1024	38.56	10.63	22.87
2048	**38.82**	**10.71**	**23.11**
3072	38.59	10.59	22.89

Table 18: Meta-review *sent-ctrl* generation results of different source truncation lengths.

labels in the control sequence for both controlled and uncontrolled extractive baselines, and sent-ctrl is used for all controlled extractive baselines. We also adopt the same k for the generic baselines.

For the Transformers, we first load the pretrained model and then fine-tune it on MReD. All experiments are conducted on single V100 GPUs, using a batch size of 1 in order to fit the large pretrained model on a single GPU. During fine-tuning, we set the Transformers' hyperparameters of "minimum_target_length" to 20, and "maximum_target_length" to 400, according to our filter range on the meta-review lengths. For the rest of the hyperparameters, we use the pretrained model's default values. Due to long inputs (see Table 17), we experiment with different source truncation lengths of 1024, 2048, and 3072 tokens. Due to the limitation of GPU space, we cannot explore truncation length of more than 3072 tokens.

D.5 Ablation on truncation length

By default, the Transformers truncate the source to 1024 tokens. We further investigate the performance of different source truncation lengths using *rate-concat*. As shown in Table 18, truncating the source to 2048 tokens consistently achieves the best performance.

D.6 Attention aggregation method

During generation, we can obtain the attention weights of each output token towards all input tokens. Specifically, we average all decoder layers' cross attention weights for the same output token generated at each decoding step. We then calculate an attention value for that output token on each input sentence, by aggregating the token's attention weights on the list of input tokens that belong to the same sentence by max pooling. Finally, we can calculate an output-sentence-to-input-sentence attention score, by adding up these attention values for the output tokens that belong to the same sentence.

Common attention aggregation methods include summation, average-pooling, and max-pooling. We use max-pooling to aggregate attention for same-sentence input tokens, because summation gives high attention scores to excessively long sentences due to attention weight accumulation, whereas average-pooling disfavors long sentences containing a few relevant phrases by averaging the weights out. With max-pooling, we can correctly identify sentences with spiked attention at important phrases, regardless of sentence lengths. For attention aggregation on the same-sentence output tokens, summation is used and can be viewed as allowing each output token to vote an attention score on all input sentences, so that the input sentence receiving the highest total score is the most relevant. We conduct trial runs of all aggregation methods on input tokens with summation for output-token aggregation for multiple generation examples, and indeed max-pooling outperforms the other two by identifying more relevant input sentences with the generated sentence.

Once we have the attention scores, we can attribute the generation of each output sentence to a few topmost relevant input sentences. Then, we can draw a color map of the input tokens in the selected sentences based on their relative attention weights.

D.7 Structure-controlled generation examples

We show examples of the generation results using alternative control sequences on another submission in Table 16. We can see the effectiveness of controlling the output structure using our proposed method.

D.8 Human evaluation

For structure similarity, we instruct the judges to label each generated sentence with the closest category. We then calculate the normalized token-level edit distance between the judge-annotated label sequence and the given control sequence, then deduct this value from 1.

For decision correctness, we evaluate it on a binary scale where 1 indicates complete correctness and 0 otherwise. More specifically, we give 0 if the generation produces contradictory decisions and a wrong decision, or the generation does not show enough hints for rejection or acceptance.