Improving Iterative Text Revision by Learning Where to Edit from Other Revision Tasks

Zae Myung Kim[1,3*] , **Wanyu Du**[2] , **Vipul Raheja**[3] , **Dhruv Kumar**[3] , **Dongyeop Kang**[1]
[1]University of Minnesota, [2]University of Virginia, [3]Grammarly
{kim01756,dongyeop}@umn.edu wd5jq@virginia.edu,
{vipul.raheja,dhruv.kumar}@grammarly.com

Abstract

Iterative text revision improves text quality by fixing grammatical errors, rephrasing for better readability or contextual appropriateness, or reorganizing sentence structures throughout a document. Most recent research has focused on understanding and classifying different types of edits in the iterative revision process from human-written text instead of building accurate and robust systems for iterative text revision. In this work, we aim to build an end-to-end text revision system that can iteratively generate helpful edits by explicitly detecting *editable spans* (where-to-edit) with their corresponding edit intents and then instructing a revision model to revise the detected edit spans. Leveraging datasets from other related text editing NLP tasks, combined with the specification of editable spans, leads our system to more accurately model the process of iterative text refinement, as evidenced by empirical results and human evaluations. Our system significantly outperforms previous baselines on our text revision tasks and other standard text revision tasks, including grammatical error correction, text simplification, sentence fusion, and style transfer. Through extensive qualitative and quantitative analysis, we make vital connections between edit intentions and writing quality, and better computational modeling of iterative text revisions.

1 Introduction

Text revision, naturally, is an iterative process. Writers are required to simultaneously and repeatedly comprehend multiple requirements, such as covering the content, and following linguistic norms and discourse conventions, when producing well-written texts (Flower, 1980; Collins and Gentner, 1980; Vaughan and McDonald, 1986). Most recent text editing studies have either focused on general-purpose text revision (Malmi et al., 2019; Mallinson et al., 2020; Stahlberg and Kumar, 2020; Li et al., 2022), or targeted monolingual sequence transduction tasks individually, such as grammatical error correction (GEC) (Awasthi et al., 2019; Omelianchuk et al., 2020; Chen et al., 2020), text simplification (Dong et al., 2019; Kumar et al., 2020; Omelianchuk et al., 2021; Agrawal et al., 2021), and text style transfer (Madaan et al., 2020; Malmi et al., 2020; Reid and Zhong, 2021), among others.

	Tag	Gran.	Iter.
LaserTagger (Malmi et al., 2019)	O	S	×
FELIX (Mallinson et al., 2020)	O	S&P	×
Seq2Edits (Stahlberg and Kumar, 2020)	O	S	×
IteraTeR (Du et al., 2022b)	×	S&P	✓
DeLIteraTeR (Ours)	I	S&P	✓

Table 1: Comparison with previous works. Gran. for Granularity: S for sentence-level and P for paragraph-level. Iter. for Iterativeness. Tag for the type of Edit Tagging: O for Edit Operations, I for Edit Intentions.

Despite their progress, these works are quite restricted in their generalizability to practical use cases: (1) They generally rely on learning edit *operations*, such as `ADD`, `KEEP`, `DELETE`, and `REPLACE`, which fail to account for many nuanced edit operations such as complex phrasal or sentence rewrites such as word reordering (Malmi et al., 2019) or other complex paragraph-level edits. A fundamental limitation of these surface-level edit operations is that they fail to capture the underlying intentions behind the resulting edit operations, and hence, do not learn anything about *why* a part of text was edited in a certain way. For example, a certain span of words may be replaced because it is unclear (CLARITY) or disfluent (FLUENCY); and depending on this edit *intent* (as opposed to superficial edit *operations*), the revised outcome could be different (Section 5.2). (2) These tagging approaches are inherently limited as they have been developed for *sentence*-level editing (Mallinson

[*]The work was done while Zae Myung Kim was interning at Grammarly.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 9986 - 9999
December 7-11, 2022

et al., 2020). (3) Since most of the aforementioned studies re-purpose existing sentence-level text editing tasks into monolingual tasks, they are unable to understand or reason about the *iterative* nature of revision, which more closely reflects the human revision process. Table 1 summarizes the comparison with previous works.

In this work, we propose DELITERATER: A DEL*ineate*-E*dit*-I*terate* approach for the task of Iterative Text Revision (Du et al., 2022b). Our approach is composed of three stages: (1) *Delineate*: We first detect *editable spans*, the spans of text that require edits, along with their desired edit intentions such as coherence and fluency using a span detection model. (2) *Edit*: A text revision model then generates the revised text conditioned on the detected *editable spans*. (3) *Iterate*: The system then continues to iteratively revise the text by going back to Stage 1 (Delineate) until it does not generate further edits or reaches a predefined maximum revision depth.

The main difference of DELITERATER from ITERATER (Du et al., 2022b) is that the editable spans are detected first before starting surface-level revisions, making revisions more interpretable and controllable. Also, each editable span is grounded on corresponding edit intentions, providing more nuanced reasoning behind the edit operations. We also extend Du et al. (2022b)'s ITERATER dataset (we refer to the augmented dataset as ITERATER+) by incorporating data from other text editing tasks, leading to significant improvements in performance.[1]

Our method shows significant improvements on the Iterative Text Editing task, as well as four well-established monolingual text editing tasks: GEC, sentence fusion, split & rephrase, text simplification, and formality style transfer.

2 Related Work

Our work is most closely related to Du et al. (2022b), who formally introduced the task of Iterative Text Revision by releasing an annotated dataset of iteratively revised texts, and also used it to provide edit suggestions in a human-in-the-loop iterative editing setting (Du et al., 2022a). In both their works, they computationally model the iterative text revision process, leveraging edit intent information by simply appending it to the input text. However, we improve on their modeling formulation, as evidenced by our experimental results in a significant way: instead of simply appending edit intentions at the beginning of any sentence, we provide more fine-grained edit intention information to our text revision model by first detecting the exact spans which require an edit. Moreover, by incorporating edit-intention-specific knowledge from external task-specific datasets, we are able to push the performance further.

3 DELITERATER

We follow the Iterative Text Revision task as introduced by Du et al. (2022b): given a source document $\mathcal{D}^{t-1}$, at each revision depth t, a text revision system will apply a set of edits to get the revised document $\mathcal{D}^{t}$. The system will continue iterating revision until the revised document $\mathcal{D}^{t}$ satisfies a set of predefined stopping criteria, such as reaching a predefined maximum revision depth t_{max}, or making no edits between $\mathcal{D}^{t-1}$ and $\mathcal{D}^{t}$.

In this section, we describe ITERATER+, the augmented version of the iterative text revision dataset (Du et al., 2022b) and the system pipeline for DELITERATER.

3.1 ITERATER+: Augmented Dataset

We use the Iterative Text Revision dataset (ITERATER) released by Du et al. (2022a,b) as our primary dataset. Under their dataset taxonomy, each text edit is broadly categorized into one of two groups: MEANING-CHANGED and NON-MEANING-CHANGED. Further, edits that belong to the latter group are further assigned to one of the following five sub-groups: FLUENCY, COHERENCE, CLARITY, STYLE, and OTHER. This taxonomy of *edit intents* reflects writers' general "intention" when revising formal documents. It allows us to model the purpose behind each edit of texts, providing more in-depth information than just superficial *edit actions* such as `ADD`, `KEEP`, and `DELETE`.

In this work, we build upon the ITERATER dataset by gathering data from other similar text editing tasks according to the aforementioned taxonomy as ITERATER (Section 3.1.2).

3.1.1 Pre-processing

We observe that many edits from the original dataset were actually MEANING-CHANGED edits, i.e., the revised text embodied significantly different content from the old text. This often oc-

[1]The datasets, codes, and models can be found at https://github.com/vipulraheja/iterater.

Edit Intention	Dataset	Example Input	Example Output
FLUENCY	NUCLE 2014	Technology based on scientific research requires a wide range of knowledge about the research.	Technology based on scientific research requires a wide range of knowledge to conduct the research.
	Lang-8	These days, I write my daily schedule on a notebook.	These days, I write my daily schedule in a notebook.
COHERENCE	DiscoFuse	Their flight is weak. They run quickly through the tree canopy.	Their flight is weak, but they run quickly through the tree canopy.
CLARITY	NEWSELA	A storm surge is what forecasters consider a hurricane's most treacherous aspect.	A storm surge is considered a hurricane's most dangerous aspect.
	WikiLarge	Wyolica is a two-piece group from Japan.	Wyolica is a two person band from Japan.
	Split and Rephrase	Aaron Deer plays guitar in Indie rock style whose origins are coming from the new wave music.	Aaron Deer is a an Indie rock guitar player. The stylistic origin of indie rock is new wave music.
STYLE	GYAFC	They wouldnt want u stepping in.	They would not desire your interference.

Table 2: Examples of data instances from external corpora used to create the augmented ITERATER+ dataset.

curred when the revisions were made at a document level, reorganizing the paragraphs while adding new contents. In addition, there were also many cases where significant amounts of texts were either added or deleted by the revisions. Since new content generation is not the scope of our task, we filtered these type of edits by comparing *length ratio* and *character-level similarity* between original and revised strings, discarding nearly 40% of the dataset.

3.1.2 Data Augmentation

The ITERATER dataset taxonomy is general enough to encompass other text editing tasks. For example, the datasets from the GEC task can be viewed as datasets for FLUENCY, text simplification task as CLARITY edits, sentence fusion or splitting as COHERENCE, and formality style transfer as STYLE. Using this insight, we adopted the following external datasets for our system. These datasets underwent an identical pre-processing routine as the main dataset.

Fluency We use two prominent corpora for GEC: the NUS Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013), which consists of 1,414 essays written by students at the National University of Singapore (NUS); and the NAIST Lang-8 Corpus of Learner English (Tajiri et al., 2012), which is one of the largest and most widely used datasets for GEC. The essays in the NUCLE Corpus were responses to some prompts from various topics including technology innovation and health care, and were hand-corrected by professional English instructors. Lang-8 Corpus, on the other hand, was created by language learners correcting each other's texts. Although these datasets contain multiple fine-grained error categories specific to GEC, in this work, we consider all errors in these corpora as FLUENCY, following the comprehensive definition of fluency in the ITERATER dataset taxonomy.

Clarity We use the Newsela corpus (Xu et al., 2015) for the Text Simplification task, which consists of 1,130 articles and their simplified versions which were created by professional editors at Newsela, an online education platform. We also use WikiLarge, another benchmark dataset for the text simplification task. It was constructed from automatically-aligned complex-simple sentence pairs from English Wikipedia and Simple English Wikipedia (Zhu et al., 2010; Woodsend and Lapata, 2011; Kauchak, 2013). We use the standardized split of this dataset released by Zhang and Lapata (2017) consisting of 296k complex-simple sentence pairs. Finally, we use the Split and Rephrase (Narayan et al., 2017) dataset, which includes 1.06M instances mapping a single complex sentence to a sequence of sentences that express the same meaning. We labeled the edits collected from these datasets as CLARITY edits.

Coherence We use the DiscoFuse dataset (Geva et al., 2019), which provides a large collection of pairs of sentences that were originally from one coherent sentence, and segmented into two by a rule-based method from sports articles and Wikipedia. The task then involves linking these two sentences as coherently as possible where it could be done through inserting a discourse connective or merg-

Intentions	Dataset	Sentences	Edits
FLUENCY	ITERATER	131k	131k
	TASK-SPECIFIC	124k	162k
CLARITY	ITERATER	109k	109k
	TASK-SPECIFIC	22k	28k
COHERENCE	ITERATER	28k	26k
	TASK-SPECIFIC	133k	145k
STYLE	ITERATER	3k	3k
	TASK-SPECIFIC	45k	90k

Table 3: Data statistics of ITERATER+. Table 8 contains full data statistics.

ing the input sentences, etc. These dataset samples were labeled as COHERENCE for our work.

Style We use Grammarly's Yahoo Answers Formality Corpus (GYAFC) (Rao and Tetreault, 2018) which contains 110k informal and formal sentence pairs. We note that the notion of STYLE edits can be quite subjective as it is about conveying writers' writing preferences, including emotions, tone, and voice; and informal to formal conversion can be viewed as one aspect of STYLE. We use the dataset for learning informal to formal rewriting because ITERATER dataset has primarily been developed for mostly formal writing domains such as ArXiv, Wikipedia and News.

Table 3 shows the statistics and intent distributions of all datasets after the pre-processing routine. Table 2 depicts instances of data points from all of the external corpora mentioned in this section.

3.2 System Pipeline

Our system is arranged in a pipeline where edit intent classification is conducted at token-level as a structured prediction task, followed by span-based text revision where the predicted intent labels are inserted as tag spans in the input. Figure 1 highlights the overall process of the pipeline at a given revision depth, with an illustrative example. In the iterative text revision setting, this illustrated process repeats until either no editable spans are detected, or a predefined maximum revision depth is reached.

3.2.1 Intent Span Detection

Rather than predicting a single edit intent at input-level as done in Du et al. (2022a), our model predicts intents at token-level. Training of such a model was difficult without the construction of our new dataset, since the original ITERATER dataset contained a lot of noisy revisions that caused the token-level model to be degenerate.

The intent classification model was trained by fine-tuning a token-level classification layer on top of the pre-trained ROBERTA-LARGE model (Liu et al., 2019), where the input to the model is a plain text and output is one of the five classes (CLARITY, COHERENCE, FLUENCY, STYLE, NONE) for every token in the input. We also experimented with a multi-task learning by adding an input-level binary classification layer that predicts if the entire input needs revisions or not.

3.2.2 Span-Based Text Revision

The token-level predictions of the intent span prediction model are turned into intent-annotated spans where the part of text that is predicted to be edited is surrounded with intent tags as shown in Figure 1. This way, the revision model can focus on parts of the input that need revisions. Note that it is possible to have multiple intent spans in which case the model revises multiple parts of the input.

Following Du et al. (2022b), we also fine-tune a PEGASUS model (Zhang et al., 2020) which is a Transformer-based (Vaswani et al., 2017) sequence-to-sequence (Seq2Seq) model. While more lightweight non-autoregressive models such as FELIX (Mallinson et al., 2020) are available, we opted for using the autoregressive Seq2Seq models as our main choice of models. This is because the task of generating text from editable spans is more involved than generating text from edit operations, showing better performance as described in Du et al. (2022b).

4 Quantitative Results

While the recent previous studies on text revision include FELIX (Mallinson et al., 2020), LaserTagger (Malmi et al., 2019), Seq2Edits (Stahlberg and Kumar, 2020), and ITERATER Du et al. (2022b), we mainly compare our system against the latest work, ITERATER, as it had shown consistent improvements over the other aforementioned systems in revision quality (Du et al., 2022b).

We evaluate our system, DELITERATER, on the test splits of ITERATER+ dataset which consists of the original ITERATER set and newly augmented task-specific datasets from four different text editing NLP tasks: text simplification, sentence fusion & splitting, GEC, and formality style transfer, as described in Section 3.1.2.

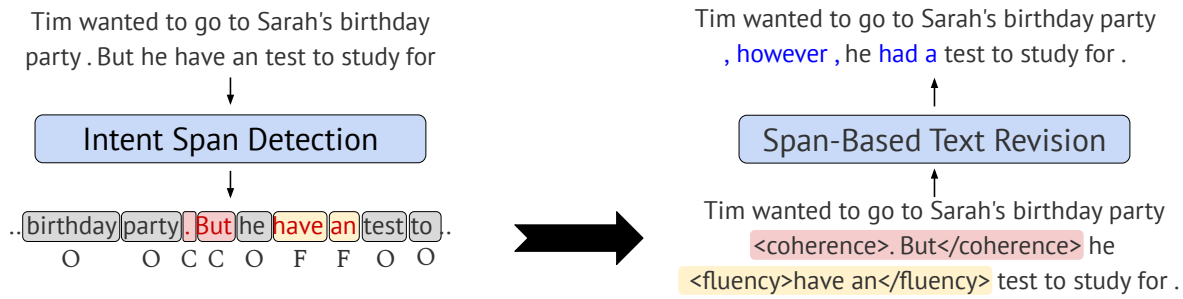


Figure 1: Illustration of the DELITERATER pipeline at a given revision depth. While the figure depicts two sentences, the pipeline works for entire paragraphs.

	Training Dataset	External Tasks (DELITERATER-*)				ITERATER-TEST				
		Clarity	Coherence	Fluency	Style	Clarity	Coherence	Fluency	Style	Overall
SS	ITERATER+	**49.87**	**98.06**	78.27	**71.89**	**34.08**	**24.44**	63.57	**0.42**	45.99
SM	ITERATER+	34.23	95.40	77.72	21.81	14.39	17.08	58.47	0.00	32.27
MS	ITERATER+	43.09	97.90	**79.27**	65.60	33.80	22.43	**67.36**	0.00	**49.13**
MM	ITERATER+	43.34	96.80	77.37	71.29	33.98	20.53	64.84	0.00	47.26

Table 4: Performance of models on intent span detection. All the models are named using the XY convention, where X refers to the Single-sentence (S) vs. Multi-sentence (M) setting and Y refers to the Single-task (S) vs. Multi-task (M) training setting.

	CLARITY-TEST			COHERENCE-TEST			FLUENCY-TEST			STYLE-TEST			ITERATER-TEST		
	B	R	S	B	R	S	B	R	S	B	R	S	B	R	S
No Edits Baseline	0.59	74.44	23.60	0.84	97.13	31.30	0.75	88.43	25.96	0.28	61.26	15.51	0.86	91.80	29.88
ITERATER-SINGLE	0.59	74.29	27.50	0.71	89.14	33.79	0.76	88.23	36.39	0.29	61.34	18.90	0.84	91.96	35.62
ITERATER-MULTI	0.59	74.29	27.50	0.71	89.14	33.79	0.76	88.23	36.39	0.29	61.34	18.90	0.87	93.19	43.22
DELITERATER-CLARITY	0.62	75.93	36.63	0.34	62.79	22.60	0.25	52.14	27.06	0.04	23.01	28.28	0.55	72.14	26.43
DELITERATER-COHERENCE	0.17	40.52	26.33	0.96	98.74	81.17	0.24	52.20	26.72	0.03	23.79	29.81	0.39	59.74	22.60
DELITERATER-FLUENCY	0.16	40.29	25.93	0.52	77.23	29.09	0.86	92.38	70.83	0.04	25.08	27.66	0.61	75.15	35.18
DELITERATER-STYLE	0.33	59.82	29.93	0.40	72.31	26.82	0.42	73.63	35.31	0.42	68.35	51.60	0.35	65.34	24.53
DELITERATER-ITERATER	0.60	75.95	51.48	0.85	95.58	51.08	0.83	90.44	61.49	0.28	56.36	36.99	**0.92**	**96.18**	62.54
DELITERATER-SINGLE	0.65	79.05	57.48	0.96	98.66	80.81	**0.87**	92.98	73.06	0.48	71.72	60.45	**0.92**	96.14	62.06
DELITERATER-MULTI	**0.66**	**79.36**	**58.70**	**0.96**	**98.73**	**81.23**	**0.87**	**93.10**	**73.95**	**0.49**	**72.13**	**61.44**	**0.92**	96.13	**64.09**

Table 5: Comparison of end-to-end Iterative Text Revision models. B is BLEU, R is ROUGE-L, and S is SARI.

4.1 Intent Span Detection

We hypothesize that the prediction of edit intention for a span may benefit from the use of information needed to predict whether an edit is needed or not. For instance, the prediction of a fluency edit may benefit from the detection of a grammatical error in the text. With this idea, we train a ROBERTA-LARGE model with two different settings:

1. *Single-Task*: trained for token-level edit intention classification.
2. *Multi-Task*: Different task-specific heads trained for each task (binary classification task of edit detection, and multi-class edit intention classification).

We also experimented with varying lengths of context windows for the edit intent prediction:

1. *Single-Sentence*: Only the tokens belonging to a sentence are classified at a given time.
2. *Multi-Sentence*: For a given sentence for which the intent span detection needs to happen, we concatenate the preceding and succeeding sentence before and after it respectively, to provide additional context to the classification model.

Table 4 shows a breakdown of the models' performance on all combinations of the single/multi-sentence and single/multi-task settings for the intent span detection task. We report F1 scores on

both the ITERATER dataset, and the test splits of TASK-SPECIFIC external datasets which we incorporated into ITERATER+. We find that the model with multi-sentence, single-task (MS) setting is the best performing one overall. We use this model as our main intent span detection model. In general, we find that multi-sentence models (MM, MS) perform better than single-sentence models (SS, SM). This can be attributed to the fact that multi-sentence models have access to more context in their inputs, and are able to leverage that to predict edits more accurately. We do notice that the single-sentence single-task (SS) model performs better on the task-specific test sets, and that is attributed to the fact that these datasets are all sentence-level, and did not contain multiple sentences for any added context.

4.2 Span-Based Text Revision

As mentioned in Section 3.2.2, our main model (DELITERATER) for span-based text revision was trained from PEGASUS-LARGE model, following Du et al. (2022b). While we do not bring in additional modeling components to the PEGASUS model, we emphasize that the successful training of the model and its performance were heavily dependent on how we constructed the inputs to the model, i.e, sentence-level vs. token-level (span-based) intent annotation. Specifically, the input to the baseline model, ITERATER, was prepared by adding a sentence-level intent class at the beginning of the sentence, whereas DELITERATER takes inputs with intent information annotated as tags within corresponding parts of the inputs as shown in Fig. 1. In addition, similar to the intent span detection (Section 4.1), we experimented with both single-sentence (SINGLE) and multi-sentence (MULTI) settings.

In Table 5, we report the performance of each model using three automatic metrics: BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), and SARI (Xu et al., 2016). We observe that span-based DELITERATER models outperform the sentence-level ITERATER models on all test sets. We also see that the *-MULTI models are generally better than *-SINGLE models, and this difference is more prominent in ITERATER-test[2] as the test set can cater for multi-sentence inputs while other task-specific sets do not.

[2]We note that all of the test sets that we used are filtered as specified in Sec. 3.1.1

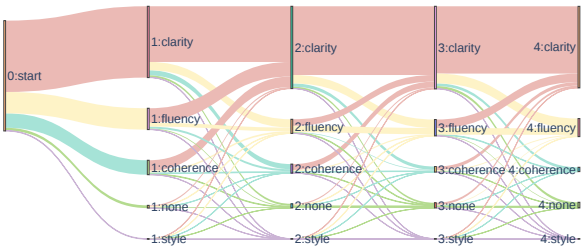


Figure 2: Illustration of intention trajectories by DELITERATER for iterative revision on ITERATER-TEST.

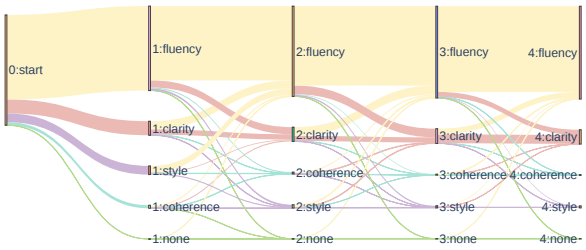
We also note that the performance of models was greatly influenced by the ITERATER+ dataset that we collected and pre-processed. To this end, we report the results of an ablation study on different portions of the dataset in the mid-section of Table 5, where DELITERATER-{CLARITY, COHERENCE, FLUENCY, STYLE} models were trained on each task-specific dataset only, while DELITERATER-ITERATER model was trained on the filtered version of ITERATER dataset. The results show that while the individual task-specific models perform well on their corresponding test sets, the quality of revision drops significantly when tested on the ITERATER-test. Similarly, the model that was solely trained on ITERATER dataset does not perform well on the task-specific test sets. As expected, the best performance was achieved when the models were trained on the full ITERATER+ dataset.

5 Qualitative Results

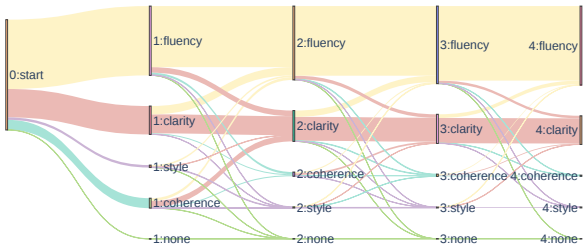
In this section, we analyze the behavior of our DELITERATER system in greater detail by looking at the development of edit intent trajectories as we run the system iteratively using its previous revision $\mathcal{D}^{t-1}$ to generate the revision at depth t (Section 5.1). In Section 5.2, we try to probe our text revision model by modifying the outputs of the span-based model to see if we can generate diverse revisions by placing (1) different intents on the same edit spans, and (2) same intent on different edit spans.

5.1 Edit intention trajectories by depths

To understand if there is a clear pattern in trajectories of edit intents with the progression of revision, Sankey diagrams were drawn for revision results ($t < 5$) on each test set. At revision depth t, we record the number of instances of all possible intent transitions, $\text{INTENT}_i^{t-1} \rightarrow \text{INTENT}_j^{t}$, defining the flow for the Sankey diagram.



(a) Essays written by low level of English skills



(b) Essays written by high level of English skills

Figure 3: Sankey diagrams on ETS Corpus of Non-Native Written English.

Revision Input	Revision Output
I **<fluency>** disagree about that "young people do not give enough time to helping their communities" **</fluency>**.	I disagree with the statement that "young people do not give enough time to helping their communities".
I **<clarity>** disagree about that "young people do not give enough time to helping their communities" **</clarity>**.	I disagree that young people do not give enough time to helping their communities.
I **<coherence>** disagree about that "young people do not give enough time to helping their communities" **</coherence>**.	I disagree.
I **<style>** disagree about that "young people do not give enough time to helping their communities" **</style>**.	I disagree with the statement that "young people do not give enough time to helping their communities".

Table 6: Effect of detected edit intentions on generated revisions.

Revision Input	Revision Output
I <fluency>disagree about that "young people do not give enough time to helping their communities"</fluency>.	I disagree with the statement that "young people do not give enough time to helping their communities".
I <fluency>disagree about</fluency> that "young people do not give enough time to helping their communities".	I disagree with that "young people do not give enough time to helping their communities".
I disagree about that "young people <clarity>do not give enough time</clarity> to helping their communities".	I disagree about that "young people do not have enough time to helping their communities".
I disagree about that "young people do not give enough time <clarity>to helping their communities</clarity>".	I disagree about that "young people do not give enough time to help their communities."

Table 7: Effect of detected edit spans (for the same intents) on generated revisions.

Figure 2 shows a Sankey diagram for ITERATER-test dataset.[3] On this test set, we can observe that most of the intent sequences are flowing from CLARITY to CLARITY, followed by edit sequences that go into FLUENCY. The diagram is, of course, influenced by the distribution of edit intents present in the test set. With the task-specific test sets, we confirm that the corresponding intents tend to be the major flow in the diagrams. This begs the question: would we observe similar results with documents from the same domain but written by writers with different levels of English proficiency?

To answer this question, we performed a similar experiment using English essays from the ETS Corpus of Non-Native Written English (Blanchard, Daniel et al., 2014) where writers with different levels of English proficiency answered prompt questions by composing short essays. Using the validation and test sets of the corpus, we gathered 261 essays separately from two groups: (a) writers with low English proficiency vs. (b) high English proficiency. These essays answer 8 different essay topics, and the number of essays for each topic is kept the same between the groups.

Figure 3 shows the Sankey diagrams for both groups, computed using their essays and the corresponding iterative revisions. In the figure, we can identify some distinctive patterns in the flows of edit intents; for example, more FLUENCY edit transitions occur for group (a) than (b) which can be considered as more superficial errors than other intent types. For group (b), we see more COHERENCE errors that later make transition to CLARITY.

This finding suggests that there could be an optimal path of edit transition for a given document of a particular domain; which could be learned by a reinforcement learning algorithm where a reward would be a combination of automatic scores of revised documents: something we would like to explore in future work.

So far, at the current state of the models, we do sometimes observe some pitfalls where continuously applying the revision model multiple times either degrades the quality of the text by removing

[3]The rest of the diagrams are added to the Appendix C

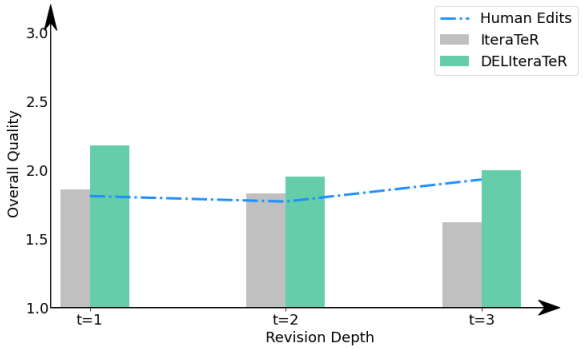


Figure 4: Manual pair-wise comparison for document revisions without MEANING-CHANGED edits.

words or gets stuck in a deadlock position where the model oscillates between applying and reverting a single revision. The former can be mitigated by stopping the revision process when a score from an automatic metric such as SARI decreases. The latter deadlock case can be filtered heuristically as well. We point out that these phenomena are also observed in other text revision models, especially more so when their training datasets include a lot of meaning-changed edits that are deletions.

5.2 Effect of intents and spans on revisions

We also try to probe the system to understand the behaviors of the constituent systems. In particular, the sensitivity of the revision generation model to the outputs of the intent span detection model. To do this, we conducted two analyses: (1) We modified the edit intentions keeping the editable span the same; and (2) while keeping the predicted edit intention the same, we varied the editable span. By modifying the inputs to the text revision model in this manner, we analyzed its outputs.

In Table 6, we can observe that placing a FLUENCY intent span revises the less fluent original sentence by linking a correct preposition for the verb "disagree" and adding the following noun, "statement". Similarly, a CLARITY intent simplifies the sentence by merging the quoted segment. However, as shown with the results from COHERENCE and STYLE intents in the table, the revised outputs may not always be preferable.

Table 7 shows the effect of varying the length of editable spans. We see that if there is a change in the original sentence (i.e. a revision is predicted), that change only occurs within the bounds of the editable span. The results obtained with the editable intent spans suggest that we can control and influence the model's generations to a certain extent.

6 Human Evaluation

To better understand how our system affects the text quality and the iterative revision process, we conducted human evaluations to investigate how do text editing models affect document quality.

We hired a group of proficient linguists to evaluate the quality of the documents being edited across multiple (up to 3) revisions, where each revision was annotated by 3 linguists. For each revision, we randomly shuffle the original and revised texts, and ask the evaluators to select which one was better in terms of fluency, coherence, readability, meaning preservation, and overall quality. They could choose one of the two texts, or neither. Then, we calculated the score for the quality of the human revisions as follows: 1 means the revised text is worse compared to the original text; 2 means the revised text does not show a better quality than the original text, or there was no agreement among the 3 annotators; 3 means that the revised text was better than the original text.

Figure 4 shows the results of the human evaluation on the aforementioned criteria. We choose our best-performing model (DELITERATER-MULTI) trained on ITERATER+ using the *delineate-edit-iterate* approach to generate revisions by first identifying editable spans, and compare with human revisions and the text revision model from Du et al. (2022b). We see that our system produces the best overall results, outperforming the human edits, as well as ITERATER system in overall quality. This is a major improvement relative to (Du et al., 2022b) where model revisions were significantly underperformed by human edits in overall quality. Table 9 shows an example of iterative text revision generated by ITERATER and DELITERATER, respectively.

7 Conclusion and Discussions

We propose DELITERATER: an improved system for Iterative Text Revision, using a *delineate-edit-iterate* framework, consisting of an intent span detection model, and a text revision generation model, based on the ITERATER framework of Du et al. (2022b). The edit intent detection model is a token-level edit-intention classification model which detects *editable spans*: spans of text that require an edit along with the type of edit needed. The text revision model is a generative model, which makes re-

visions to the detected editable spans, conditioned on their corresponding edit intentions. We also create IteraTeR+: an expanded version of (Du et al., 2022b) IteraTeR dataset by incorporating data from other text editing NLP tasks. Leveraging this dataset and the *delineate-edit-iterate* framework, our system supervises the revision generation model to reflect both the location, and intentions behind the desired revision, leading to superior performance on the Iterative Text Revision task, compared to other baselines and related works.

Experiments on the standard IteraTeR test dataset, as well as standard NLP text editing datasets demonstrate the effectiveness of our framework for the task. Moreover, human evaluations indicate that our system produces the best overall results, outperforming the human edits, as well as IteraTeR system across all revision depths. Additionally, we provide insights into our models by probing them, and into the progression of iterative text revision by analyzing the edit intent trajectories across both our test datasets as well as a dataset of English essays from the ETS corpus, hinting the possibility of learning optimal revision paths possibly thorough reinforcement learning. In the future, we plan to investigate in this direction as well as improving the general robustness of the system by task-specific data augmentation with induced noise.

8 Limitations

We note that the augmented task-specific datasets were only available at sentence-level. While the augmentation did improve the models' performance on IteraTeR-test set, it still lacked the contextual information to to unlock the full potential of multi-sentence modeling. Also, while we conducted user studies on the quality of the generated revisions, our current version of work does not yet provide results obtained with *human-in-the-loop* deployment where users are involved in the iterative revision process along with the revision system. Another limitation of our work is that the revision system is geared toward generating formal writing than informal and casual writing.

9 Ethical Considerations

All the data collected in this work is from publicly available sources, and the original document authors' copyrights are respected. During the data annotation process, all human evaluators are anonymized to respect their privacy rights. All human evaluators get a fair wage that is higher than the minimum wage based on the number of data points they evaluate. There is no risk that the harms of our work will disproportionately fall on marginalized or vulnerable populations. Our datasets do not contain any identity characteristics (e.g. gender, race, ethnicity), and will not have ethical implications of categorizing people.

In terms of our models, we recognize that by using text generation models as part of our system, they are susceptible to issues of hallucination and other potentially harmful content (Maynez et al., 2020; Gehman et al., 2020). However, since the focus of our system is on text editing, we are able to mitigate some of these issues by carefully curating our datasets. We ignore any data points which lead to meaning-changing edits, thereby reducing the chances of hallucination, or generation of new and potentially harmful content.

References

Sweta Agrawal, Weijia Xu, and Marine Carpuat. 2021. A non-autoregressive edit-based approach to controllable text simplification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3757–3769, Online. Association for Computational Linguistics.

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. 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 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Blanchard, Daniel, Tetreault, Joel, Higgins, Derrick, Cahill, Aoife, and Chodorow, Martin. 2014. Ets corpus of non-native written english.

Mengyun Chen, Tao Ge, Xingxing Zhang, Furu Wei, and Ming Zhou. 2020. Improving the efficiency of grammatical error correction with erroneous span detection and correction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7162–7169, Online. Association for Computational Linguistics.

Allan Collins and Dedre Gentner. 1980. A framework for a cognitive theory of writing. In *Cognitive processes in writing*, pages 51–72. Erlbaum.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In

Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

Wanyu Du, Zae Myung Kim, Vipul Raheja, Dhruv Kumar, and Dongyeop Kang. 2022a. Read, revise, repeat: A system demonstration for human-in-the-loop iterative text revision. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, pages 96–108, Dublin, Ireland. Association for Computational Linguistics.

Wanyu Du, Vipul Raheja, Dhruv Kumar, Zae Myung Kim, Melissa Lopez, and Dongyeop Kang. 2022b. Understanding iterative revision from human-written text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3573–3590, Dublin, Ireland. Association for Computational Linguistics.

Linda Flower. 1980. The dynamics of composing: Making plans and juggling constraints. *Cognitive processes in writing*, pages 31–50.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Mor Geva, Eric Malmi, Idan Szpektor, and Jonathan Berant. 2019. DiscoFuse: A large-scale dataset for discourse-based sentence fusion. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3443–3455, Minneapolis, Minnesota. Association for Computational Linguistics.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1537–1546, Sofia, Bulgaria. Association for Computational Linguistics.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7918–7928, Online. Association for Computational Linguistics.

Jingjing Li, Zichao Li, Tao Ge, Irwin King, and Michael Lyu. 2022. Text revision by on-the-fly representation optimization. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, pages 58–59, Dublin, Ireland. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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 preprint arXiv:1907.11692*.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. 2020. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1869–1881, Online. Association for Computational Linguistics.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. FELIX: Flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255, Online. 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.

Eric Malmi, Aliaksei Severyn, and Sascha Rothe. 2020. Unsupervised text style transfer with padded masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8671–8680, Online. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 606–616, Copenhagen, Denmark. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vipul Raheja, and Oleksandr Skurzhanskyi. 2021. Text Simplification by Tagging. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–25, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. 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 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Machel Reid and Victor Zhong. 2021. LEWIS: Levenshtein editing for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3932–3944, Online. Association for Computational Linguistics.

Felix Stahlberg and Shankar Kumar. 2020. Seq2Edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online. Association for Computational Linguistics.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for ESL learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Marie M. Vaughan and David D. McDonald. 1986. A model of revision in natural language generation. In *24th Annual Meeting of the Association for Computational Linguistics*, pages 90–96, New York, New York, USA. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing Statistical Machine Translation for Text Simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.

A Statistics on ITERATER+ Dataset

In Table 8 we show detailed statistics of our ITERATER+ dataset, showing before and after the filtering process.

Split	Intent	Source	Before		After	
			Sentences	Edits	Sentences	Edits
Train	FLUENCY	ITERATER	142*k*	142*k*	107*k*	107*k*
		TASK-SPECIFIC	1.1M	779*k*	122*k*	158*k*
	CLARITY	ITERATER	140*k*	140*k*	90*k*	90*k*
		TASK-SPECIFIC	1.69M	6.07M	22*k*	28*k*
	COHERENCE	ITERATER	70*k*	70*k*	24*k*	24*k*
		TASK-SPECIFIC	4.49M	5.1M	120*k*	132*k*
	STYLE	ITERATER	3*k*	3*k*	2.5*k*	2.5*k*
		TASK-SPECIFIC	104*k*	246*k*	29*k*	60*k*
Valid	FLUENCY	ITERATER	17*k*	17*k*	11*k*	11*k*
		TASK-SPECIFIC	-	-	1*k*	2k
	CLARITY	ITERATER	15*k*	15*k*	9*k*	9.5*k*
		TASK-SPECIFIC	42*k*	139*k*	122	170
	COHERENCE	ITERATER	8.7*k*	8.7*k*	2.5*k*	2.5*k*
		TASK-SPECIFIC	46*k*	52*k*	1*k*	1.1*k*
	STYLE	ITERATER	366	366	265	265
		TASK-SPECIFIC	41*k*	92*k*	11*k*	20*k*
Test	FLUENCY	ITERATER	20*k*	20*k*	13*k*	13*k*
		TASK-SPECIFIC	-	-	1.5*k*	1.9*k*
	CLARITY	ITERATER	17*k*	17*k*	10*k*	10*k*
		TASK-SPECIFIC	45*k*	149*k*	132	200
	COHERENCE	ITERATER	10*k*	10*k*	2*k*	2*k*
		TASK-SPECIFIC	44*k*	50*k*	12*k*	13*k*
	STYLE	ITERATER	447	447	330	330
		TASK-SPECIFIC	19*k*	45*k*	5*k*	10*k*

Table 8: Dataset splits and sizes. "Before" refers to the raw data statistics before the pre-processing routine (Section 3.1.1), and "After" refers to the data statistics after the pre-processing was applied.

B Training Details

Throughout our experiments, we mostly adopted codes released by Du et al. (2022a,b). We did not conduct any additional hyper-parameter search, but followed the same hyper-parameter settings as Du et al. (2022a,b) when training both intent classification and text revision models. We plan to make our version of codes and final datasets publicly available upon acceptance.

We used Transformers library (Wolf et al., 2020) from Hugging Face to train and run the models using four NVIDIA V100 GPUs in a distributed data-parallel setting. The intent classification models (ROBERTA-LARGE) were generally trained to convergence within 10 hours, while the text revision models (PEGASUS-LARGE) took up to five days to train for 3 epochs using our full ITERATER+ dataset. We note that we did not introduce any extra model parameters, and therefore the network sizes are identical to that of the corresponding original models. The models were saved every 2,000 batch steps and selected based on the validation performance on the ITERATER dataset.

C Edit Intention Trajectories

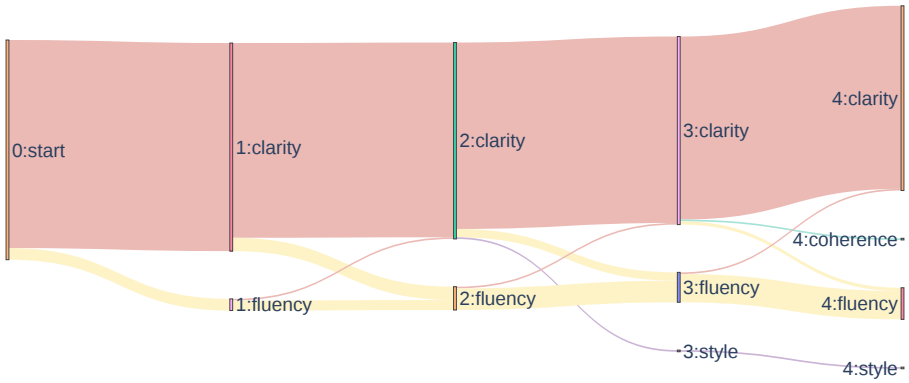
Figure 5 shows Sankey diagrams drawn for the task-specific test sets of ITERATER+ dataset.

D Document-level Revision Examples

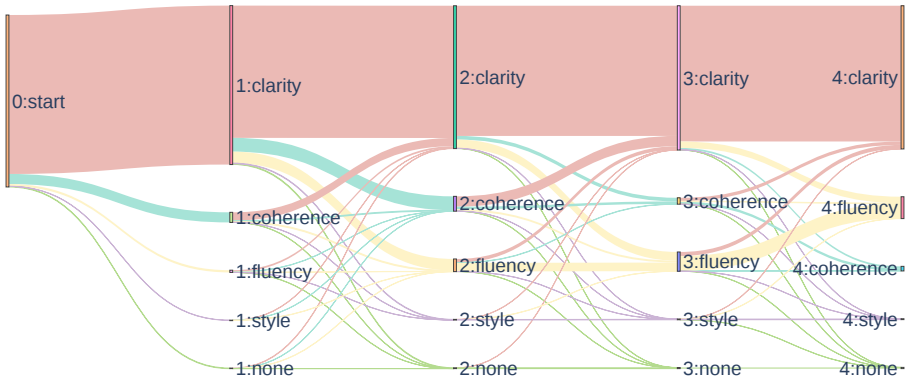
Table 9 shows iterative revisions generated by ITERATER and DELITERATER (ours), respectively.

t	Human Edits	IteraTeR	**DElIteraTeR** (ours)
0	The insurgent maoists would be allowed to return to the government. The treaty establishes that a constitutional asembly should form in April to rewrite the constitution, formally end the monarchy and put together the details of the new republican system. The maoist rebels declared a cease fire and signed a peace treaty, agreeing to place its troops and weapons under U.N. supervision.	The insurgent maoists would be allowed to return to the government. The treaty establishes that a constitutional asembly should form in April to rewrite the constitution, formally end the monarchy and put together the details of the new republican system. The maoist rebels declared a cease fire and signed a peace treaty, agreeing to place its troops and weapons under U.N. supervision.	The insurgent maoists would be allowed to return to the government. The treaty establishes that a constitutional asembly should form in April to rewrite the constitution, formally end the monarchy and put together the details of the new republican system. The maoist rebels declared a cease fire and signed a peace treaty, agreeing to place its troops and weapons under U.N. supervision.
1	The insurgent maoists would be allowed to return to the government. The treaty establishes that a constitutional asembly should form in April to rewrite the constitution, formally end the monarchy and put together the details of the new republican system. The maoist rebels declared a cease fire and signed a peace treaty, agreeing to place its troops and weapons under ~~U.N.~~ United Nations supervision.	The ~~insurgent maoists~~ insurgents would be allowed to return to the government. The treaty establishes that a constitutional asembly should ~~form~~ be formed in April to rewrite the constitution, formally end the monarchy and put together the details of the new republican system. The ~~maoist~~ Maoist rebels declared a cease fire and signed a peace treaty, agreeing to place its troops and weapons under U.N. supervision.	The ~~insurgent maoists~~ insurgents would be allowed to return to the government. ~~The treaty~~ It establishes that a constitutional asembly should form in April to rewrite the constitution, formally end the monarchy and put together the details of the new republican system. ~~The maoist rebels~~ Rebels declared a cease fire and signed a peace treaty, agreeing to place its troops and weapons under U.N. supervision.
2	The insurgent ~~maoists~~ Maoists would be allowed to return to the government. The treaty establishes that a constitutional ~~asembly~~ assembly should form in April to rewrite the constitution, formally end the monarchy and put together the details of the new republican system. The ~~maoists~~ Maoists rebels declared a cease fire and signed a peace treaty, agreeing to place its troops and weapons under United Nations supervision.	The insurgents would be allowed to return to the government. The treaty establishes that a constitutional ~~asembly~~ assembly should be formed in April to rewrite the constitution, formally end the monarchy and put together the details of the new republican system. The Maoist rebels declared a cease fire and signed a peace treaty, agreeing to place its troops and weapons under U.N. supervision.	The insurgents would be allowed to return to the government. ~~It establishes that a~~ A constitutional asembly should form in April to rewrite the constitution, formally end the monarchy and put together the details of the new republican system. ~~Rebels~~ The rebels declared a cease fire and signed a peace treaty, agreeing to place its troops and weapons under U.N. supervision.
3	-	-	The insurgents would be allowed to return to the government. ~~A constitutional asembly~~ Constitutional Assembly should form in April to rewrite the constitution, formally end the monarchy and put together the details of the new republican system. ~~The rebels~~ Rebels declared a cease fire and signed a peace treaty, agreeing to place its troops and weapons under U.N. supervision.

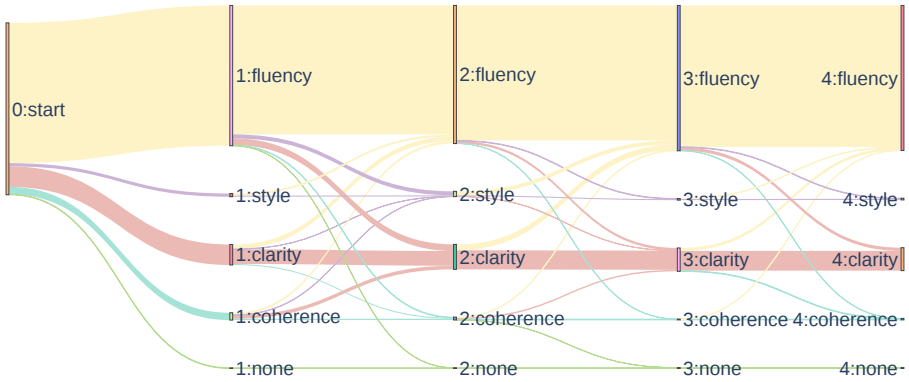
Table 9: A sample snippet of iterative text revisions generated by human writer, IteraTeR and DElIteraTeR (ours), where t is the revision depth and $t = 0$ indicates the original input text. Note that ~~text~~ represents deletions, and text represents insertions.



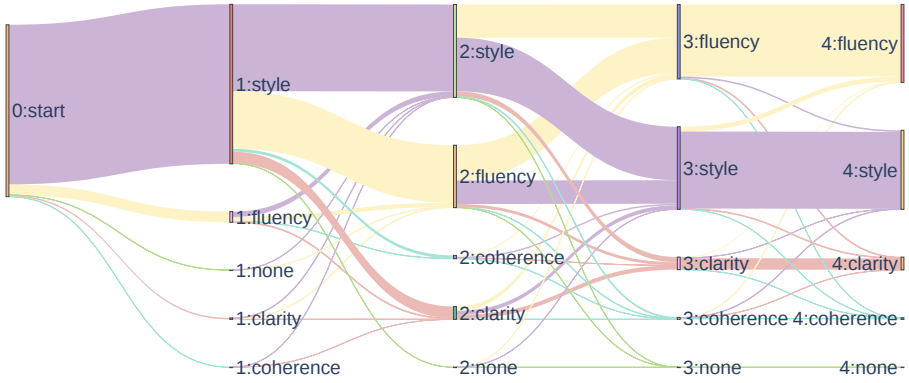
(a) CLARITY



(b) COHERENCE



(c) FLUENCY



(d) STYLE

Figure 5: Sankey diagrams illustrating edit intention trajectories for task-specific test sets