Understanding Iterative Revision from Human-Written Text

Wanyu Du[1]*, **Vipul Raheja**[2] , **Dhruv Kumar**[2] , **Zae Myung Kim**[3] ,
Melissa Lopez[2] , **Dongyeop Kang**[4]
[1]University of Virginia, [2]Grammarly,
[3]Univ. Grenoble Alpes, CNRS, LIG, [4]University of Minnesota
wd5jq@virginia.edu
{vipul.raheja,dhruv.kumar,melissa.lopez}@grammarly.com
zae-myung.kim@univ-grenoble-alpes.fr
dongyeop@umn.edu

Abstract

Writing is, by nature, a strategic, adaptive, and more importantly, an iterative process. A crucial part of writing is editing and revising the text. Previous works on text revision have focused on defining edit intention taxonomies within a single domain or developing computational models with a single level of edit granularity, such as sentence-level edits, which differ from human's revision cycles. This work describes ITERATER: the first large-scale, multi-domain, edit-intention annotated corpus of iteratively revised text. In particular, ITERATER is collected based on a new framework to comprehensively model the iterative text revisions that generalize to various domains of formal writing, edit intentions, revision depths, and granularities. When we incorporate our annotated edit intentions, both generative and edit-based text revision models significantly improve automatic evaluations.[1] Through our work, we better understand the text revision process, making vital connections between edit intentions and writing quality, enabling the creation of diverse corpora to support computational modeling of iterative text revisions.

1 Introduction

Writing is a complex and effortful cognitive task, where writers balance and orchestrate three distinct cognitive processes: planning, translation, and revising (Flower and Hayes, 1980). These processes can be hierarchical and recursive and can occur at any moment during writing. This work focuses on text revision as an essential part of writing (Scardamalia, 1986). Revising text is a strategic, and adaptive process. It enables writers to deliberate over and organize their thoughts, find a better line of argument, learn afresh, and discover what was not known before (Sommers, 1980). Specifically, text revision involves identifying discrepancies between intended and instantiated text, deciding what edits to make, and how to make those desired edits (Faigley and Witte, 1981; Fitzgerald, 1987; Bridwell, 1980).

Text revision is an *iterative* process. Human writers are unable to simultaneously comprehend multiple demands and constraints of the task when producing well-written texts (Flower, 1980; Collins and Gentner, 1980; Vaughan and McDonald, 1986) – for instance, expressing ideas, covering the content, following linguistic norms and discourse conventions of written prose, etc. Thus, they turn towards making *successive iterations of revisions* to reduce the number of considerations at each time.

Previous works on iterative text revision have three major limitations: (1) simplifying the task to an noniterative "original-to-final" text paraphras-

Each comment was annotated by three different annotators, which achieved high inter-annotator agreement. The proposed annotation {~~process~~ approach} CLARITY is also language and domain independent{, nevertheless, it was currently applied for Brazilian Portuguese} MEANING-CHANGED .
Each comment was annotated by three different annotators, {~~which~~ and} COHERENCE achieved high inter-annotator agreement. The {new} MEANING-CHANGED proposed annotation approach is also language and {~~domain independent, nevertheless, it was currently~~domain-independent (although it has been} CLARITY applied for Brazilian Portuguese{)} FLUENCY .
Each comment was annotated by three different annotators {~~,~~} FLUENCY and achieved high inter-annotator agreement. The {~~new~~} COHERENCE proposed annotation approach is also language and domain-independent {~~(although it has been applied~~ nevertheless it is currently customized} COHERENCE for Brazilian Portuguese {~~)~~} FLUENCY .

Table 1: An iteratively revised ArXiv abstract snippet (2103.14972, version 2, 3, and 4) with our annotated EDIT-INTENTION in ITERATER.

*This research was performed when Wanyu Du was interning at Grammarly.

[1]Code and dataset are available at https://github.com/vipulraheja/IteraTeR.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 3573 - 3590

ing; (2) focusing largely on sentence-level editing (Faruqui et al., 2018; Botha et al., 2018; Ito et al., 2019; Faltings et al., 2021); (3) developing editing taxonomies within individual domains (e.g. Wikipedia articles, academic writings) (Yang et al., 2017; Zhang et al., 2017; Anthonio et al., 2020). These limitations make their proposed text editing taxonomies, datasets, and models lose their generalizability and practicality.

We present ITERATER— an annotated dataset for ITERAtive TExt Revision that consists of 31,631 iterative document revisions with sentence-level and paragraph-level edits across multiple domains of formally human-written text, including Wikipedia[2], ArXiv[3] and Wikinews.[4] Table 1 shows a sample ArXiv document in ITERATER, that underwent iterative revisions. Our dataset includes 4K manually annotated and 196K automatically annotated edit intentions based on a sound taxonomy we developed, and is generally applicable across multiple domains and granularities (See Table 2). Note that ITERATER is currently only intended to support formal writing revisions, as iterative revisions are more prevalent in formal rather than informal writings (e.g. tweets, chit-chats)[5]. Our contributions are as follows:

- formulate the iterative text revision task in a more comprehensive way, capturing greater real-world challenges such as successive revisions, multi-granularity edits, and domain shifts.
- collect and release a large, multi-domain Iterative Text Revision dataset: ITERATER, which contains 31K document revisions from Wikipedia, ArXiv and Wikinews, and 4K edit actions with high-quality edit intention annotations.
- analyze how text quality evolves across iterations and how it is affected by different kinds of edits.
- show that incorporating the annotated edit-intentions is advantageous for text revision systems to generate better-revised documents.

2 Related Work

Edit Intention Identification. Identification of edit intentions is an integral part of the iterative text revision task. Prior works have studied the categorization of different types of edit actions to help understand why editors do what they do and how effective their actions are (Yang et al., 2017; Zhang et al., 2017; Ito et al., 2019). However, these works do not further explore how to leverage edit intentions to generate better-revised documents. Moreover, some of their proposed edit intention taxonomies are constructed with a focus on specific domains of writing, such as Wikipedia articles (Anthonio et al., 2020; Bhat et al., 2020; Faltings et al., 2021) or academic essays (Zhang et al., 2017). As a result, their ability to generalize to other domains remains an open question.

Dataset	Size	Domain	Gran.	Hist.	Ann.
Yang et al. (2017)	5K	Wiki	P	×	√
Anthonio et al. (2020)	2.7M	Wiki	S	√	×
Zhang et al. (2017)	180	Academic	S	√	√
Spangher and May (2021)	4.6M	News	S	√	×
ITERATER (Ours)	31K	All	S&P	√	√

Table 2: Comparisons with previous related works. Gran. for Granularity: S for sentence-level and P for paragraph-level. Hist. for Revision History. Ann. for Edit Intention Annotations.

Noniterative Text Revision Models. Some prior works (Faruqui et al., 2018; Botha et al., 2018; Ito et al., 2019; Faltings et al., 2021) simplify the text revision task to a single-pass "original-to-final" sentence-to-sentence generation task. However, it is very challenging to conduct multiple perfect edits at once. For example, adding transition words or reordering the sentences are required to further improve the document quality. Therefore, single-pass sentence-to-sentence text revision models are not sufficient to deal with real-world challenges of text revision tasks. In this work, we explore the performance of text revision models in multiple iterations and multiple granularities.

Iterative Text Revision Datasets. While some prior works have constructed iterative text revision datasets, they are limited to singular writing domains, such as Wikipedia-style articles (Anthonio et al., 2020), academic essays (Zhang et al., 2017) or news articles (Spangher and May, 2021). In this work, we develop a unified taxonomy to analyze the characteristics of iterative text revision behaviors across different domains and collect large scale text revisions of human writings from multiple domains. The differences between ITERATER and the prior datasets are summarized in Table 2.

[2] https://www.wikipedia.org/
[3] https://arxiv.org/
[4] https://www.wikinews.org/
[5] Further extension to less formal writings (e.g. blog, emails) will be discussed in the future.

	ITERATER-FULL						ITERATER-HUMAN					
	ArXiv		Wikipedia		Wikinews		ArXiv		Wikipedia		Wikinews	
Depth	#D	#E	#D	#E	#D	#E	#D	#E	#D	#E	#D	#E
1	9,446	65,450	8,195	51,290	7,878	39891	95	618	130	1,072	173	1,227
2	1,615	11,391	1,991	12,868	1,455	8,116	76	499	38	250	25	155
3	301	2,076	415	2,786	161	1,704	6	47	10	98	4	27
4	66	444	64	723	16	71	1	13	1	12	0	0
5	15	107	9	52	4	18	0	0	0	0	0	0
Total	11,443	79,468	10,674	67,719	9,514	49,800	178	1,177	179	1,432	202	1,409

Table 3: Statistics of the ITERATER dataset, where #D indicate the number of document revisions ($\mathcal{R}^t$), and #E indicate the number of annotated edit actions.

3 Formulation: Iterative Text Revision

We provide formal definitions of the Iterative Text Revision task, and its building blocks.

Edit Action. An edit action $\boldsymbol{a}_k$ is a local change applied to a certain text object, where k is the index of the current edit action. The local changes include: insert, delete and modify. The text objects include: token, phrase[6], sentence, and paragraph. This work defines local changes applied to tokens or phrases as *sentence-level edits*, local changes applied to sentences as *paragraph-level edits* and local changes applied to paragraphs as *document-level edits*.

Edit Intention. An edit intention $\boldsymbol{e}_k$ reflects the revising goal of the editor when making a certain edit action. In this work, we assume each edit action $\boldsymbol{a}_k$ will only be labeled with one edit intention $\boldsymbol{e}_k$. We further describe our edit intention taxonomy in Table 4 and §4.2.1.

Document Revision. A document revision is created when an editor saves changes for the current document (Yang et al., 2016, 2017). One revision $\mathcal{R}^t$ is aligned with a pair of documents $(\mathcal{D}^{t-1}, \mathcal{D}^t)$ and contains K^t edit actions, where t indicates the version of the document and $K^t \geq 1$. A revision with K^t edit actions will correspondingly have K^t edit intentions:

$$(\mathcal{D}^{t-1}, \mathcal{D}^t) \rightarrow \mathcal{R}^t = \{(\boldsymbol{a}_k^t, \boldsymbol{e}_k^t)\}_{k=1}^{K^t} \quad (1)$$

We define t as the revision depth.

Iterative Text Revision. Given a source text $\mathcal{D}^{t-1}$, iterative text revision is the task of generating revisions of text $\mathcal{D}^t$ at depth t until the quality of the text in the final revision satisfies a set of pre-defined stopping criteria $\{s_0, ..., s_M\}$:

$$\mathcal{D}^{t-1} \xrightarrow{g(\mathcal{D})} \mathcal{D}^t, \text{if } f(\mathcal{D}^t) < \{s_0, ..., s_M\} \quad (2)$$

where $g(\mathcal{D})$ is a text revision system and $f(\mathcal{D})$ is a quality evaluator of the revised text. The quality evaluator $f(\mathcal{D})$ can be automatic systems or manual judgements which measure the quality of the revised text. The stop criteria $\{s_i\}$ is a set of conditions that determine whether to continue revising or not. In this work, we simply set them as revision depth equal to 10, and edit distance between $\mathcal{D}^{t-1}$ and $\mathcal{D}^t$ equal to 0 (§6.2). We will include other criteria which measures the overall quality, content preservation, fluency, coherence and readability of the revised text in future works.

4 ITERATER Dataset

4.1 Raw Data Collection

Domains. We select three domains – Wikipedia articles, academic papers, and news articles – to cover different human writing goals, formats, revision patterns, and quality standards. The three domains consist of formally written texts, typically edited by multiple authors. We describe why and how we collect text revision from each domain below:

- **Scientific Papers.** Scientific articles are written in a rigorous, logical manner. Authors generally highlight and revise their hypotheses, experimental results, and research insights in this domain. We collect paper abstracts submitted at different timestamps (i.e., version labels) from ArXiv.
- **Wikipedia Articles.** Encyclopedic articles are written in a formal, coherent manner, where editors typically focus on improving the clarity and structure of articles to make people easily understand all kinds of factual and abstract encyclope-

[6] In this work, we define phrase as text pieces which contain more than one token and only appears within a sentence.

Edit-Intention	Description	Example	Counts (Ratio)
FLUENCY	Fix grammatical errors in the text.	She went to the ~~markt~~market.	942 (23.44%)
COHERENCE	Make the text more cohesive, logically linked and consistent as a whole.	She works hard~~. She~~; therefore, she is successful.	393 (9.78%)
CLARITY	Make the text more formal, concise, readable and understandable.	The changes ~~made the paper better than before~~improved the paper.	1,601 (39.85%)
STYLE	Convey the writer's writing preferences, including emotions, tone, voice, etc..	Everything was awfully rotten.	128 (3.19%)
MEANING-CHANGED	Update or add new information to the text.	This method improves the model accuracy from 64% to ~~78~~83%.	896 (22.30%)
OTHER	Edits that are not recognizable and do not belong to the above intentions.	This method is also named as ~~CITATION1~~.	58 (1.44%)

Table 4: A taxonomy of edit intentions in ITERATER, where FLUENCY, COHERENCE, CLARITY and STYLE belong to NON-MEANING-CHANGED edits.

dic information. We collect revision histories of the main contents of Wikipedia articles.

- **News Articles.** News articles are generally written in a precise and condensed way. News editors emphasize improving the clarity and readability of news articles to keep people updated on rapidly changing news events. We collect revision histories of news content from Wikinews.

Raw Data Processing. We first collect all raw documents, then sort each document version according to its timestamp in ascending order. For each document $\mathcal{D}$, we pair two consecutive versions as one revision $(\mathcal{D}^{t-1}, \mathcal{D}^t) \rightarrow \mathcal{R}^t$, where t is the revision depth. For each sampled document-revision $\mathcal{R}^t$, we extract its full edit actions using *latexdiff*.[7] We provide both the paragraph-level and sentence-level revisions where the latter is constructed by applying a sentence segmentation tool,[8] and aligning each sentence to each revision. For each revision pair, we have: the revision type, the document id, the revision depth, an original phrase and a revised phrase, respectively.[9] The detailed processing of raw text is described in Appendix A.

In summary, we collect 31,631 document revisions with 196,987 edit actions, and maintain a relatively balanced distribution across three domains, as shown in Table 3. We call this large-scale dataset as ITERATER-FULL-RAW.

[7] https://www.ctan.org/pkg/latexdiff

[8] https://github.com/zaemyung/sentsplit

[9] We also record character-level indices of their positions within the original sentence and the paragraph.

4.2 Data Annotation

To better understand the human revision process, we sample 559 document revisions from ITERATER-FULL-RAW, consisting of 4,018 human edit actions. We refer to this small-scale unannotated dataset as ITERATER-HUMAN-RAW. In §4.2.2, we then use Amazon Mechanical Turk (AMT) to crowdsource edit intention annotations for each edit action according to our proposed edit-intention taxonomy (§4.2.1). We refer to this small-scale annotated dataset as ITERATER-HUMAN.[10]

We then scale these manual annotations to ITERATER-FULL-RAW by training edit intention prediction models on ITERATER-HUMAN, and automatically label ITERATER-FULL-RAW to construct ITERATER-FULL. (§4.2.3)

4.2.1 Edit Intention Taxonomy

For manual annotations, we propose a new edit intention taxonomy in ITERATER (Table 4), in order to comprehensively model the iterative text revision process. Our taxonomy builds on prior literature (Rathjens, 1985; Harris, 2017). At the highest level, we categorize the edit intentions into ones that change the meaning or the information contained in the text (MEANING-CHANGED), and ones that preserve these characteristics (NON-MEANING-CHANGED). Since our goal is to understand edit intentions to improve the quality of writing, we focus on categorizing edits in the latter category further into four sub-categories: FLUENCY, CLARITY, COHERENCE and STYLE. Our proposed taxonomy of edit intentions is generally applicable to multiple

[10] We provide our annotation instruction in Appendix C.

Edit-Intention	Precision	Recall	F1
CLARITY	0.75	0.63	0.69
FLUENCY	0.74	0.86	0.80
COHERENCE	0.29	0.36	0.32
STYLE	1.00	0.07	0.13
MEANING-CHANGED	0.44	0.69	0.53

Table 5: Edit intention classifier performance on the test split of ITERATER-HUMAN.

domains, edit-action granularities (sentence-level and paragraph-level), and revision depths. We also propose the OTHER category for edits that cannot be labeled using the above taxonomy.

4.2.2 Manual Annotation

Since edit intention annotation is a challenging task, we design strict qualification tests to select 11 qualified AMT annotators (details in Appendix B). To further improve the annotation quality, we ask another group of expert linguists (English L1, bachelor's or higher degree in Linguistics) to re-annotate the edits which do not have a majority vote among the AMT workers. Finally, we take the majority vote among 3 human annotations (either from AMT workers or from expert linguists) as the final edit intention labels. This represents the ITERATER-HUMAN dataset. We release both the final majority vote and the three raw human annotations per edit action as part of the dataset.

4.2.3 Automatic Annotation

To scale up the annotation, we train an edit-intention classifier to annotate ITERATER-FULL-RAW and construct the ITERATER-FULL dataset. We split the ITERATER-HUMAN dataset into 3,254/400/364 training, validation and test pairs. The edit intention classifier is a RoBERTa-based (Liu et al., 2020) multi-class classifier that predicts an intent given the original and the revised text for each edit action[11]. Table 5 shows its performance on the test set. The Fluency and Clarity edit intentions are easy to predict with F1 scores of 0.8 and 0.69, respectively, while Style and Coherence edit intentions are harder to predict with F1 scores of 0.13 and 0.32, respectively, largely due to the limited occurrence of Style and Coherence intents in the training data (Table 4).

	ArXiv	Wikipedia	Wikinews	All
1st-round	0.3369	0.3630	0.3886	0.3628
2nd-round	0.4983	0.4274	0.5601	0.5014

Table 6: Inter-annotator agreement (Fleiss' κ (Fleiss, 1971)) across two rounds of annotations, where the 1st-round only contains annotations from qualified AMT workers, and the 2nd-round contains annotations from both qualified AMT workers and expert linguists.

4.3 Data Analysis

Edit Intention Distributions. The iterative edit intention distributions in three domains are demonstrated in Figure 1. Across all three domains, authors tend to make the majority of edits at revision depth 1. However, the number of edits rapidly decreases at revision depth 2, and few edits are made at revision depth 3 and 4.

We find that CLARITY is one of the most frequent edit intentions across all domains, indicating that authors focus on improving readability across all domains. For ArXiv, MEANING-CHANGED edits are also among the most frequent edits, which indicates that authors also focus on updating the contents of their abstracts to share new research insights or update existing ones. Meanwhile, ArXiv also covers many FLUENCY and COHERENCE edits, collecting edits from scientific papers and suggesting meaningful revisions would be an important future application of our dataset. For Wikipedia, we find that FLUENCY, COHERENCE, and MEANING-CHANGED edits roughly share a similar frequency, which indicates Wikipedia articles have more complex revision patterns than ArXiv and news articles. For Wikinews, FLUENCY edits are equally emphasized, indicating that improving grammatical correctness of the news articles is just as important.

Inter-Annotator Agreement. We measure inter-annotator agreement (IAA) using the Fleiss' κ (Fleiss, 1971). Table 6 shows the IAA across three domains. After the second round of re-annotation by proficient linguists, the Fleiss' κ increases to 0.5014, which indicates moderate agreement among annotators.

We further look at the raw annotations where at least 1 out of 3 annotators assigns a different edit intention label. We find that the COHERENCE intention is the one that is the most likely to have a disagreement: 312 out of 393 COHERENCE an-

[11] Please refer to Appendix D for more training details.

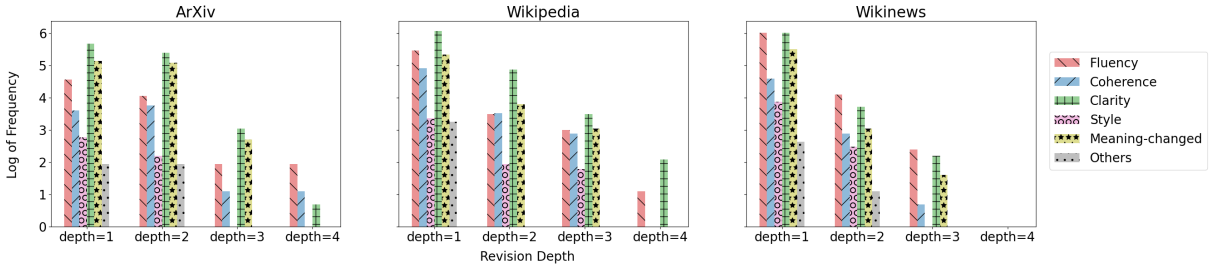


Figure 1: Logarithm (base e) of frequency for edit-intentions in each revision depth for the three dataset domains.

notations do not have consensus. Within those disagreements of the COHERENCE intention, 68.77% are considered to be CLARITY, and 11.96% are considered to be the FLUENCY intention. Annotators also often disagree on the CLARITY intention, where 1023 out of 1601 CLARITY intentions do not have a consensus. Among those disagreements of the CLARITY intention, 30.33% are considered to be COHERENCE, and 30.23% are considered to be STYLE.

The above findings explain why the inter-annotator agreement scores are lower in Wikipedia and ArXiv. As shown in Figure 1, Wikipedia has many COHERENCE edits while ArXiv has many CLARITY edits. This explains the difficulty of the edit intention annotation task: it not only asks annotators to infer the edit intention from the full document context, but also requires annotators to have a wide range of domain-specific knowledge in scientific writings.

5 Understanding Iterative Text Revisions

To better understand how text revisions affect the overall quality of documents, we conduct both manual and automatic evaluations on a sampled set of document revisions.

5.1 Experiment Setups

Evaluation Data. We sample two sets of text revisions for different evaluation purposes. The first set contains 21 iterative document revisions, consisting of 7 unique documents, each document having 3 document revisions from revision depth 1 to 3. The second set contains 120 text pairs, each associated with exactly one edit intention of FLUENCY, COHERENCE, CLARITY or STYLE. We validate the following research questions:

RQ1 How do human revisions affect the text quality across revision depths?

RQ2 How does text quality vary across edit intentions?

Manual Evaluation Configuration. We hire a group of proficient linguists to evaluate the overall quality of the documents/sentences, where each revision is annotated by 3 linguists. For each revision, we randomly shuffle the original and revised texts, and ask the evaluators to select which one has better overall quality. They can choose one of the two texts, or neither. Then, we calculate the score for the overall quality of the human revisions as follows: -1 means the revised text has worse overall quality than the original text; 0 means the revised text do not show a better overall quality than the original text, or cannot reach agreement among 3 annotators; 1 means the revised text has better overall quality than the original text.

Automatic Evaluation Configuration. We select four automatic metrics to measure the document quality on four different aspects: Syntactic Log-Odds Ratio (SLOR) (Kann et al., 2018) for text fluency evaluation, Entity Grid (EG) score (Lapata and Barzilay, 2005) for text coherence evaluation, Flesch–Kincaid Grade Level (FKGL) (Kincaid et al., 1975) for text readability evaluation and BLEURT score (Sellam et al., 2020) for content preservation evaluation. We describe the detailed justification of our metric selection in Appendix E. However, in our following experiments, we find these existing automatic metrics are poorly correlate with manual evaluations.

5.2 Quality Analyses on Revised Texts

RQ1: Iterative Revisions vs. Quality. Table 7 shows the document quality changes at different revision depths. Generally, human revisions improve the overall quality of original documents, as indicated by the overall score at each revision depth.[12] However, the overall quality keeps decreasing as the revision depth increases from 1 to 3, likely because it is more difficult for evaluators to grasp the

[12]We further validate this observation in another set of 50 single document-revisions in Appendix F.

t	Overall ↑	BLEURT↑	ΔSLOR ↑	ΔEG ↑	ΔFKGL ↓
1	0.4285	0.1982	-0.0985	-0.0132	-1.0718
2	0.4285	0.1368	-0.1025	-0.0295	-2.4973
3	0.1428	-0.0224	-0.0792	0.0278	1.8131

Table 7: Evaluation results for 21 iterative document revisions, where t indicates the revision depth. Note that ΔSLOR, ΔEG and ΔFKGL are computed by subtracting the scores of original documents from the scores of revised documents. Overall is the manual evaluation of overall quality of the revised documents.

FLUENCY	COHERENCE	CLARITY	STYLE
0.3673	0.1500	0.2800	-0.0385

Table 8: Manually evaluated text quality of 120 single sentence-level edits for different edit intentions.

overall quality in the deeper revision depths in the pair-wise comparisons between the original and revised documents, because less NON-MEANING-CHANGED edits have been conducted in deeper revision depths. For automatic metrics, we find ΔSLOR and ΔEG are not well-aligned with human overall score, we further examine whether human revisions makes original documents less fluent and less coherent in the analysis of RQ2.

RQ2: Edit Intentions vs. Quality. Table 8 shows how text quality varies across edit intentions. We find that FLUENCY and COHERENCE edits indeed improve the overall quality of original sentences according to human judgments. This finding suggests that ΔSLOR and ΔEG are not well-aligned with human judgements, and calls for the need to explore other effective automatic metrics to evaluate the fluency and coherence of revised texts. Besides, we observe that STYLE edits degrade the overall quality of original sentences. This observation also makes sense since STYLE edits reflect the writer's personal writing preferences (according to our edit intention taxonomy in Table 4), which not necessarily improve the readability, fluency or coherence of the text.

6 Modeling Iterative Text Revisions

To better understand the challenges of modeling the task of iterative text revisions, we train different types of text revision models using ITERATER.

6.1 Experiment Setups

Text Revision Models. For training the text revision models, we experiment with both edit-based and generative models. For the edit-based model, we use FELIX (Mallinson et al., 2020), and for the generative models, we use BART (Lewis et al., 2020) and PEGASUS (Zhang et al., 2020a). FELIX decomposes text revision into two sub-tasks: Tagging, which uses a pointer mechanism to select the subset of input tokens and their order; and Insertion, which uses a masked language model to fill in missing tokens in the output not present in the input. BART and PEGASUS are Transformer-based encoder-decoder models which are used in a wide range of downstream tasks such as natural language inference, question answering, and summarization.

Model	Dataset	SARI	BLEU	R-L	Avg.
FELIX	HUMAN-RAW	29.23	49.48	63.43	47.38
FELIX	HUMAN	30.65	54.35	59.06	48.02
FELIX	FULL-RAW	30.34	55.10	56.49	47.31
FELIX	FULL	**33.48**	**61.90**	**63.72**	**53.03**
BART	HUMAN-RAW	33.20	**78.59**	85.20	65.66
BART	HUMAN	34.77	74.43	84.45	64.55
BART	FULL-RAW	33.88	78.55	86.05	66.16
BART	FULL	**37.28**	77.50	**86.14**	**66.97**
PEGASUS	HUMAN-RAW	33.09	**79.09**	86.77	66.32
PEGASUS	HUMAN	34.43	78.85	86.84	66.71
PEGASUS	FULL-RAW	34.67	78.21	**87.06**	66.65
PEGASUS	FULL	**37.11**	77.60	86.84	**67.18**
Baseline	-	29.47	81.25	88.04	66.25

Table 9: Model performances on the test set of ITERATER-HUMAN. Baseline refers to a no-edit baseline, where we simply use the input text as the output. **Avg.** is the average score of **SARI**, **BLEU** and **R-L**.

Training. We use four training configurations to evaluate whether edit intention information can help better model text revisions. The first configuration uses the pure revision pairs without edit intention annotations (ITERATER-HUMAN-RAW dataset). In the second configuration, we include the manually annotated edit intentions to the source text (ITERATER-HUMAN dataset). Similarly, for the third and fourth training configurations, we use ITERATER-FULL-RAW dataset (no edit intention information) and ITERATER-FULL dataset (automatically annotated labels, as described in §4.2.3, simply appended to the input text). We use these four configurations for all model architectures.

6.2 Results Analysis

Automatic Evaluation. Table 9 shows the results of the three models for our different training configurations. Following prior works (Malmi et al., 2019; Dong et al., 2019; Mallinson et al., 2020), we report SARI, BLEU, and ROUGE-L

	Human Revision	Tie	Model Revision
Overall	83.33%	10.00%	6.67%
Content	13.33%	70.00%	16.67%
Fluency	50.00%	50.00%	0.00%
Coherence	40.00%	56.67%	3.33%
Readability	86.67%	10.00%	3.33%

Table 10: Manual pair-wise comparison for 30 single document revisions without Meaning-changed edits.

t	Human Revisions	Tie	Model Revisions
1	57.14%	14.28%	28.58%
2	57.14%	14.28%	28.58%
3	42.85%	57.15%	0.00%

Table 11: Manual pair-wise comparison for overall quality of 21 iterative document-revisions, where t indicates the revision depth.

metrics, and include detailed breakdown of scores in Appendix H. It is noteworthy that the SARI score on the no-edit baseline is the lowest, which indicates the positive impact of revisions on document quality, as also corroborated by the human evaluations in §5. For both ITERATER-HUMAN and ITERATER-FULL datasets, we see that edit intention annotations help to improve the performance of both FELIX and PEGASUS. Also, both models perform better on the larger ITERATER-FULL dataset compared to the ITERATER-HUMAN dataset, showing that the additional data (and automatically-annotated annotations) are helpful.

Manual Evaluation. Table 10 shows how the model revision affects the quality of the original document. We choose PEGASUS trained on ITERATER-FULL to generate revisions and compare with human revisions, as the model produces the best overall results[13]. There exists a big gap between the best-performing model revisions and human revisions, indicating the challenging nature of the modeling problem. Thus, while model revisions can achieve comparable performance with human revisions on fluency, coherence and meaning preservation, human revisions still outperform in terms of readability and overall quality.

Table 11 demonstrates how model-generated text quality varies across revision depths. In the first two depths, human revisions win over model revisions with a ratio of 57.14%. However, in the last depth, model revisions stay similar with human revisions in a ratio of 57.15%. Upon reviewing revisions in the last depth, we find a lot of MEANING-CHANGED edits in human revisions. At the same time, the model revisions only made a few FLUENCY or CLARITY edits, which the human evaluators tend to judge as "tie".

[13]We provide detailed manual evaluation configuration in Appendix G.

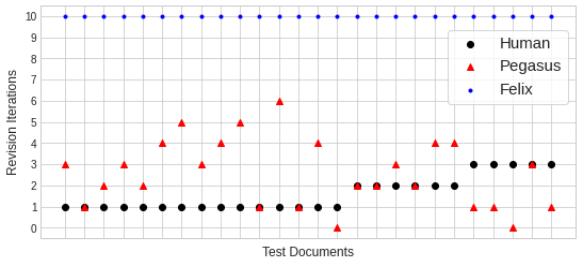


Figure 2: Number of iterations made by humans and different text revision models.

Iterativeness. We also compare the iterative ability between the two kinds of text revision models (best performing versions of both FELIX and PEGASUS: trained on ITERATER-FULL), against human's iterative revisions. Figure 2 shows that while PEGASUS is able to finish iterating after 2.57 revisions on average, FELIX continues to make iterations until the maximum cutoff of 10 that we set for the experiment. In contrast, humans on average make 1.61 iterations per document. While FELIX is able to make meaningful revisions (as evidenced by the improvements in the SARI metric in Table 14), it lacks the ability to effectively evaluate the text quality at a given revision, and decide whether or not to make further changes. PEGASUS, on the other hand, is able to pick up on these nuances of iterative revision, and learns to stop revising after a certain level of quality has been reached.

7 Conclusions and Discussions

Our work is a step toward understanding the complex process of iterative text revision from human-written texts. We collect, annotate and release ITERATER: a novel, large-scale, domain-diverse, annotated dataset of human edit actions. Our research shows that different domains of text have different distributions of edit intentions, and the general quality of the text has improved over time. Computationally modeling the human's revision process is still under-explored, yet our results indicate some interesting findings and potential directions.

Despite the deliberate design of our dataset collection, ITERATER only includes *formally* written texts. We plan to extend it to diverse sets of revi-

sion texts, such as informally written blogs and less informal but communicative texts like emails, as well as increase the size of the current dataset. For future research, we believe ITERATER can serve as a basis for future corpus development and computationally modeling iterative text revision.

8 Ethical Considerations

We collect all data from publicly available sources, and respect copyrights for original document authors. During the data annotation process, all human annotators are anonymized to respect their privacy rights. We provide fair compensation to all human annotators, where each annotator gets paid more than the minimum wage and based on the number of annotations they conducted.

Our work has no possible harms to fall disproportionately on marginalized or vulnerable populations. Our dataset does not contain any identity characteristics (e.g. gender, race, ethnicity), and will not have ethical implications of categorizing people.

Acknowledgments

We thank all linguistic expert annotators at Grammarly for annotating, evaluating and providing feedback during our data annotation and evaluation process. We appreciate that Courtney Napoles and Knar Hovakimyan at Grammarly helped coordinate the annotation resources. We also thank Yangfeng Ji at University of Virginia and the anonymous reviewers for their helpful comments.

References

Talita Anthonio, Irshad Bhat, and Michael Roth. 2020. wikiHowToImprove: A resource and analyses on edits in instructional texts. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5721–5729, Marseille, France. European Language Resources Association.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Irshad Bhat, Talita Anthonio, and Michael Roth. 2020. Towards modeling revision requirements in wikiHow instructions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8407–8414, Online. Association for Computational Linguistics.

Jan A. Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. 2018. Learning to split and rephrase from Wikipedia edit history. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 732–737, Brussels, Belgium. Association for Computational Linguistics.

Lillian S. Bridwell. 1980. Revising strategies in twelfth grade students' transactional writing. *Research in the Teaching of English*, 14(3):197–222.

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

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.

Micha Elsner and Eugene Charniak. 2008. Coreference-inspired coherence modeling. In *Proceedings of ACL-08: HLT, Short Papers*, pages 41–44, Columbus, Ohio. Association for Computational Linguistics.

Lester Faigley and Stephen Witte. 1981. Analyzing revision. *College composition and communication*, 32(4):400–414.

Felix Faltings, Michel Galley, Gerold Hintz, Chris Brockett, Chris Quirk, Jianfeng Gao, and Bill Dolan. 2021. Text editing by command. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5259–5274, Online. Association for Computational Linguistics.

Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018. WikiAtomicEdits: A multilingual corpus of Wikipedia edits for modeling language and discourse. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 305–315, Brussels, Belgium. Association for Computational Linguistics.

Jill Fitzgerald. 1987. Research on revision in writing. *Review of educational research*, 57(4):481–506.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

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

Linda Flower and John R. Hayes. 1980. The cognition of discovery: Defining a rhetorical problem. *College Composition and Communication*, 31(1):21–32.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Dynamic multi-level multi-task learning for sentence simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 462–476, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Robert A Harris. 2017. *Writing with clarity and style: A guide to rhetorical devices for contemporary writers*. Routledge.

Takumi Ito, Tatsuki Kuribayashi, Hayato Kobayashi, Ana Brassard, Masato Hagiwara, Jun Suzuki, and Kentaro Inui. 2019. Diamonds in the rough: Generating fluent sentences from early-stage drafts for academic writing assistance. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 40–53, Tokyo, Japan. Association for Computational Linguistics.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323, Brussels, Belgium. Association for Computational Linguistics.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Charles J Kowalski. 1972. On the effects of non-normality on the distribution of the sample product-moment correlation coefficient. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 21(1):1–12.

Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *IJCAI*, pages 1085–1090.

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 the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. 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. 2020. RoBERTa: A robustly optimized BERT pretraining approach. In *International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Annie Louis and Ani Nenkova. 2012. A coherence model based on syntactic patterns. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1157–1168, Jeju Island, Korea. 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.

Islam Nassar, Michelle Ananda-Rajah, and Gholamreza Haffari. 2019. Neural versus non-neural text simplification: A case study. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 172–177, Sydney, Australia. Australasian Language Technology Association.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266, Florence, Italy. Association for Computational Linguistics.

Adam Pauls and Dan Klein. 2012. Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 959–968, Jeju Island, Korea. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Dietrich Rathjens. 1985. The seven components of clarity in technical writing. *IEEE Transactions on Professional Communication*, PC-28(4):42–46.

M. Scardamalia. 1986. Research on written composition. *Handbook of reserch on teaching*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text

generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Marina Solnyshkina, Radif Zamaletdinov, Ludmila Gorodetskaya, and Azat Gabitov. 2017. Evaluating text complexity and flesch-kincaid grade level. *Journal of Social Studies Education Research*, 8(3):238–248.

Nancy Sommers. 1980. Revision strategies of student writers and experienced adult writers. *College composition and communication*, 31(4):378–388.

Radu Soricut and Daniel Marcu. 2006. Discourse generation using utility-trained coherence models. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 803–810, Sydney, Australia. Association for Computational Linguistics.

Alexander Spangher and Jonathan May. 2021. *NewsEdits*: A dataset of revision histories for news articles (technical report: Data processing). https://arxiv.org/abs/2104.09647.

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.

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.

Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2016. Who did what: Editor role identification in wikipedia. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 10.

Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2017. Identifying semantic edit intentions from revisions in Wikipedia. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2000–2010, Copenhagen, Denmark. Association for Computational Linguistics.

Fan Zhang, Homa B. Hashemi, Rebecca Hwa, and Diane Litman. 2017. A corpus of annotated revisions for studying argumentative writing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1578, Vancouver, Canada. Association for Computational Linguistics.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Daniel Zwillinger and Stephen Kokoska. 1999. *CRC standard probability and statistics tables and formulae*. CRC Press.

A Details on Text Processing in IteraTeR

For Wikipedia and Wikinews, we use the MediaWiki Action API[14] to retrieve raw pages updated at different timestamps. For each article, we start from July 2021 and trace back to its five most recent updated versions. Then, we parse[15] plain texts from raw wiki-texts and filter out all references and external links.For Wikipedia, we retrieve pages under the categories listed on the main category page [16]. For Wikinews, we retrieve pages listed on the published articles page[17].

For ArXiv, we use the ArXiv API[18] to retrieve paper abstracts. Note that we do not retrieve the full paper for two reasons: (1) some paper reserved their copyright for distribution, (2) parsing and aligning editing actions in different document types (e.g. pdf, tex) is challenging. For each paper, we start from July 2021 and retrieve all its previous submissions. We collect papers in the fields of Computer Science, Quantitative Biology, Quantitative Finance, and Economics.

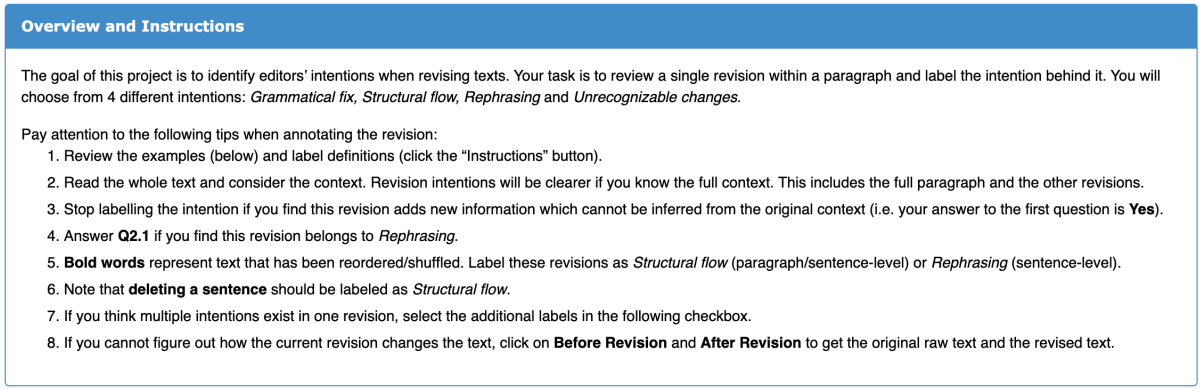
[14] https://www.mediawiki.org/wiki/API:Main_page

[15] https://github.com/earwig/mwparserfromhell

[16] https://en.wikipedia.org/wiki/Wikipedia:Contents/Categories

[17] https://en.wikinews.org/wiki/Category:Published

[18] https://arxiv.org/help/api/



Overview and Instructions

The goal of this project is to identify editors' intentions when revising texts. Your task is to review a single revision within a paragraph and label the intention behind it. You will choose from 4 different intentions: *Grammatical fix, Structural flow, Rephrasing* and *Unrecognizable changes*.

Pay attention to the following tips when annotating the revision:

1. Review the examples (below) and label definitions (click the "Instructions" button).
2. Read the whole text and consider the context. Revision intentions will be clearer if you know the full context. This includes the full paragraph and the other revisions.
3. Stop labelling the intention if you find this revision adds new information which cannot be inferred from the original context (i.e. your answer to the first question is **Yes**).
4. Answer **Q2.1** if you find this revision belongs to *Rephrasing*.
5. **Bold words** represent text that has been reordered/shuffled. Label these revisions as *Structural flow* (paragraph/sentence-level) or *Rephrasing* (sentence-level).
6. Note that **deleting a sentence** should be labeled as *Structural flow*.
7. If you think multiple intentions exist in one revision, select the additional labels in the following checkbox.
8. If you cannot figure out how the current revision changes the text, click on **Before Revision** and **After Revision** to get the original raw text and the revised text.

Figure 3: A screenshot of the annotation instruction for human annotators.

Examples

Intention	Definition	Explanation	Example
Grammatical fix	Sentence-level revisions which fix spelling and grammatical errors.	Fix spelling	She went to the {markt=>market}.
		Fix grammar error	She is planning to study ~~at~~ tonight.
		Fix grammar error	She {catched=>caught} the ball.
Structural flow	Sentence/Paragraph/Document-level revisions which make sentences logically connected and make the document more organized, including modifying transition words, splitting sentences, merging sentences, reordering sentences, etc.	Add transitions	Humpty Dumpty sat on a wall. He, therefore, had a great fall.
		Delete transitions	Smith (2019) argues for the former hypothesis. ~~Moreover,~~ Baldwin (2018) argues for the former hypothesis.
		Modify transitions	It started out quite chaotic. {Moreover=>Finally}, it all fell into place.
		Split a sentence	She worked hard throughout her early life {and, through=>.Through} patience and perseverance, became a successful lawyer.
		Merge two or more sentences	He works hard {. H=>; therefore, h}e is successful.
		Delete a sentence	~~I will not be able to meet you.~~ Sorry for missing the party. I have been busy this week.
Rephrasing	Sentence-level revisions which rewrite the text *without* adding new information.	Use less diverse vocabulary, and simpler grammar and structure	Editors help {optimize=>improve} people's communication. The changes {made the paper better than before=>improved the paper}.
		Be concise and direct, state something in the least number of words	This task can be the building block of applications ~~that requires a deep understanding of the language~~ such as fake news detection and medical claim verification.
		Change tense from past to present and vice versa	In our work, we {tried=>try} to prove a new theory. We {conducted=>conduct} massive experiments on benchmark datasets.
		Change tone from formal to informal and vice versa	Let's {meet=>chill} on Sunday afternoon.
		Change tone from neutral to emotional and vice versa	They {focus=>exclusively focus} on text-level manipulation. The food in this restaurant {sucks=>is not delicious}.

Figure 4: A screenshot of the provided examples for human annotators.

B Details on Qualificiation Tests for Human Annotation

First, we prepare a small test set with 67 edit-actions and deploy parallel test runs on AMT to get more workers participate in this task. Before starting the annotation, workers are required to pass a qualification test which has 5 test questions to get familiar with our edit-intention taxonomy. Second, we compare workers' annotations with our golden annotations, and select workers who have an accuracy over 0.4. After 5 test runs, we select 11 AMT workers who are qualified to participate in this task. Then, we deploy the full 4K edit-actions on AMT, and collect 3 human annotations per edit-action.

C Human Annotation Instruction and Interface

To guide human annotators make accurate edit-intention annotation, we provide them with a short task instruction (Figure 3) followed by some concrete edit-intention examples (Figure 4). Then, we highlight the edit-action within the document-revision and ask human annotators three questions to obtain the accurate edit-intention of the current edit-action, as illustrated in Figure 5. Note that in our previous test runs on AMT, we find that AMT workers can hardly have a consensus on Clarity and Style edits, which give a very low IAA score. Therefore, in the annotation interface, we

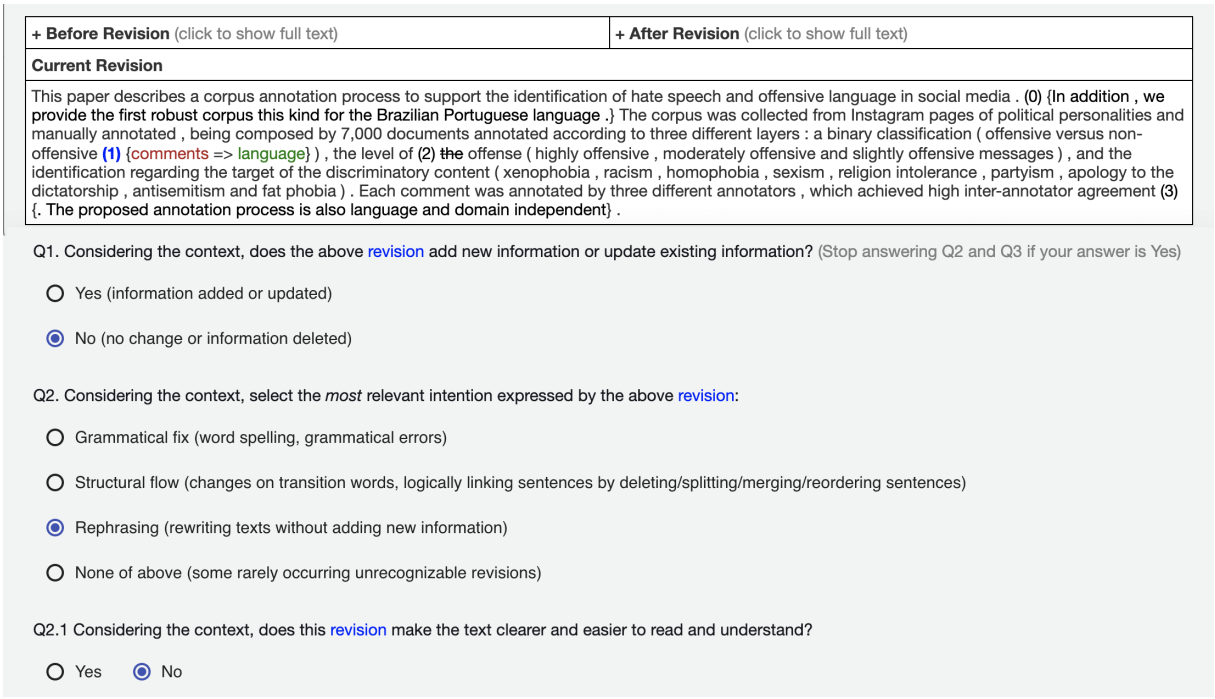


Figure 5: A screenshot of the annotation interface for human annotators.

include Clarity and Style edits under the category of "Rephrasing", and further ask the annotators to judge whether the current "Rephrasing" edit is making the text more clearer and understandable. If yes, we convert this edit to Clarity, otherwise we convert this edit to Style. This interface configuration gives us the best IAA score among our 5 test runs.

D Details on Computational Experiments

For all computational experiments in this work, we deploy them on a single Quadro RTX 4000(16GB) GPU.

RoBERTa. We leverage the RoBERTa-large model from Huggingface transformers (Wolf et al., 2020), which has 354 million parameters. We set the total training epoch to 15 and batch size to 4. We use the Adam optimizer with weight decay (Loshchilov and Hutter, 2018), and set the learning rate to 10^{-5} which decreases linearly to 0 at the last training iteration. We report descriptive statistics with a single run. We use the sklearn package (Pedregosa et al., 2011) to calculate the precision, recall and f1 score.

Text Revision Models. We leverage the BART-large (with 400 million parameters) and PEGASUS-large (with 568 million parameters) from Huggingface transformers (Wolf et al., 2020). We set the total training epoch to 5 and batch size to 16. We use the Adam optimizer with weight decay (Loshchilov and Hutter, 2018), and set the learning rate to 3×10^{-5} which decreases linearly to 0 at the last training iteration. We report descriptive statistics with a single run. We use the metrics package from Huggingface transformers to calculate the SARI, BLEU, ROUGE-1/2/L score.

E Justification of Automatic Evaluation Metrics

For **Fluency**, we use the Syntactic Log-Odds Ratio (SLOR) (Kann et al., 2018) to evaluate the naturalness and grammaticality of the current revised document, where a higher SLOR score indicates a more fluent document. Prior works (Pauls and Klein, 2012; Kann et al., 2018) found word-piece log-probability correlates well with human fluency ratings. For **Coherence**, we use the Entity Grid (EG) score (Lapata and Barzilay, 2005) to evaluate the local coherence of the current revised document, where a higher EG score indicates a more coherent document. EG is a widely adopted (Soricut and Marcu, 2006; Elsner and Charniak, 2008; Louis and Nenkova, 2012) metric for measuring document coherence. For **Readability**, we use the the Flesch–Kincaid Grade Level (FKGL) (Kincaid et al., 1975) to evaluate how easy the current re-

Overall ↑	BLEURT↑	ΔSLOR ↑	ΔEG ↑	ΔFKGL ↓
0.5800	0.4709	-0.0757	-0.0098	-0.6301

Table 12: Evaluation results for 50 document-revisions. Note that ΔSLOR, ΔEG and ΔFKGL are computed by subtracting the scores of original documents from the scores of revised documents.

vised document is for the readers to understand, where a lower FKGL indicates a more readable document. FKGL is a popular metric that has been used by many prior works (Solnyshkina et al., 2017; Xu et al., 2016; Guo et al., 2018; Nassar et al., 2019; Nishihara et al., 2019) to measure the readability of documents. For **Content Preservation**, we use the BLEURT score (Sellam et al., 2020) to measure how much content has been changed from the previous document to the current revised one, where a higher BLEURT score indicates more content has been preserved. BLEURT has been shown to correlate better with human judgments than other metrics that take semantic information into account, e.g. METEOR (Banerjee and Lavie, 2005) or BERTScore (Zhang et al., 2020b).

F Details on Human Evaluation for Single Human Revision Quality

Evaluation Data. To evaluate how do human revisions affect the text quality, we sample 50 single document-revisions, which contains 50 randomly sampled documents and each document has 1 document-revision.[19]

Result Analysis. In Table 12, we observe that human revised documents generally improve the overall quality of original documents. As for the automatic metrics, BLEURT indicates that human revisions preserve much of the content, and ΔFKGL shows that the readability of original documents improves by human revisions. However, ΔSLOR and ΔEG show a slight drop in performance. We conjecture this is because (1) ΔSLOR and ΔEG are not well-aligned with human judgements, or (2) human revisions make original documents less fluent and less coherent.

Correlation Analysis. To analyze how automatic metrics are correlated with human overall quality score, we compute the Pearson (Kowalski, 1972) and Spearman (Zwillinger and Kokoska, 1999) correlation coefficients between the automatic metrics and the human overall quality scores based on 50 single document-revisions and 21 iterative document-revisions. Table 13 shows that BLEURT and ΔFKGL are positively correlated with human overall quality score, while ΔSLOR and ΔEG are negatively correlated with human overall quality score.

[19] We exclude documents including Meaning-changed edits

	Human Overall	
	Pearson	Spearman
BLEURT	0.1139 (0.3626)	0.0756 (0.5465)
ΔSLOR	-0.1239 (0.3216)	-0.2218 (0.0734)
ΔEG	-0.1480 (0.2355)	0.0187 (0.8817)
ΔFKGL	0.1171 (0.3491)	0.2042 (0.1001)

Table 13: Correlation coefficients between human overall score and automatic metrics, where numbers in the parentheses is the p-value.

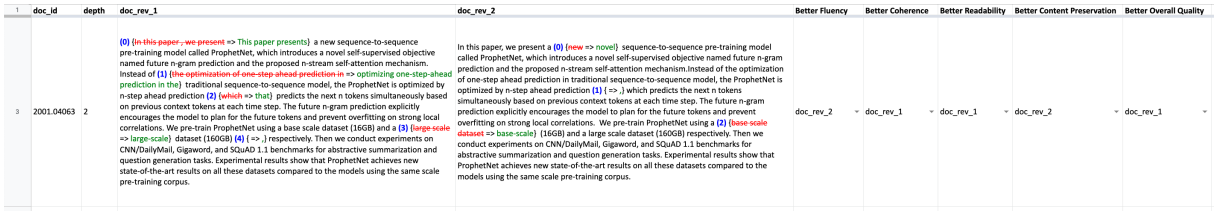
G Details on Human Evaluation Configuration for Model Revisions

First, we evaluate how do model revisions affect the quality of the document. We randomly sample 30 single document-revisions which do not contain Meaning-changed edits, and input the original documents to the best-performing model to get the model-revised documents. Then, for each data pair, we randomly shuffle model revisions and human revisions, and ask human evaluators to select which revision leads to better document quality in terms of:

- **Content Preservation**: keeping more content information unchanged;
- **Fluency**: fixing more grammatical errors or syntactic errors;
- **Coherence**: making the sentences more logically linked and organized;
- **Readability**: making the text easier to read and understand;
- **Overall Quality**: better improving the overall quality of the document.

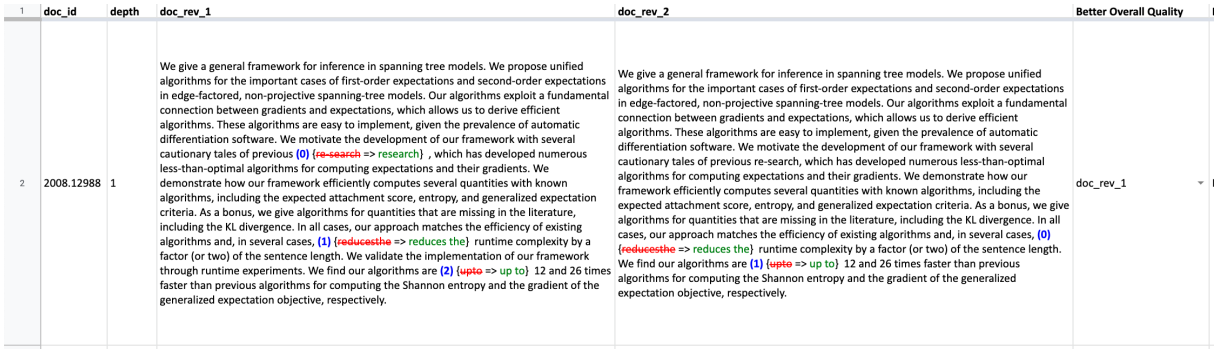
We provide the evaluation interface in Figure 6.

Secondly, we evaluate how does model generated text quality vary across revision depths. We use the same set of 21 iterative document-revisions in §5.1. We feed the original documents into the best-performing model to obtain the model revised documents at each revision depth. For each data pair, we randomly shuffle model revisions and human revisions, and ask human evaluators to judge which one gives better overall text quality. We provide the evaluation interface in Figure 7.



	doc_id	depth	doc_rev_1	doc_rev_2	Better Fluency	Better Coherence	Better Readability	Better Content Preservation	Better Overall Quality
3	2001.04063	2	(0) {In this paper , we present => This paper presents} a new sequence-to-sequence pre-training model called ProphetNet, which introduces a novel self-supervised objective named future n-gram prediction and the proposed n-stream self-attention mechanism. Instead of (1) {the optimization of one-step ahead prediction in => optimizing one-step-ahead prediction in the} traditional sequence-to-sequence model, the ProphetNet is optimized by n-step ahead prediction (2) {which => that} predicts the next n tokens simultaneously based on previous context tokens at each time step. The future n-gram prediction explicitly encourages the model to plan for the future tokens and prevent overfitting on strong local correlations. We pre-train ProphetNet using a base scale dataset (16GB) and a (3) {large scale => large-scale} dataset (160GB) (4) { => ,} respectively. Then we conduct experiments on CNN/DailyMail, Gigaword, and SQuAD 1.1 benchmarks for abstractive summarization and question generation tasks. Experimental results show that ProphetNet achieves new state-of-the-art results on all these datasets compared to the models using the same scale pre-training corpus.	In this paper, we present a (0) {new => novel} sequence-to-sequence pre-training model called ProphetNet, which introduces a novel self-supervised objective named future n-gram prediction and the proposed n-stream self-attention mechanism.Instead of the optimization of one-step ahead prediction in traditional sequence-to-sequence model, the ProphetNet is optimized by n-step ahead prediction (1) { => ,} which predicts the next n tokens simultaneously based on previous context tokens at each time step. The future n-gram prediction explicitly encourages the model to plan for the future tokens and prevent overfitting on strong local correlations. We pre-train ProphetNet using a (2) {base scale dataset => base-scale} (16GB) and a large scale dataset (160GB) respectively. Then we conduct experiments on CNN/DailyMail, Gigaword, and SQuAD 1.1 benchmarks for abstractive summarization and question generation tasks. Experimental results show that ProphetNet achieves new state-of-the-art results on all these datasets compared to the models using the same scale pre-training corpus.	doc_rev_2	doc_rev_1	doc_rev_1	doc_rev_2	doc_rev_1

Figure 6: A screenshot of the single document-revision quality evaluation interface for human evaluators.



	doc_id	depth	doc_rev_1	doc_rev_2	Better Overall Quality
2	2008.12988	1	We give a general framework for inference in spanning tree models. We propose unified algorithms for the important cases of first-order expectations and second-order expectations in edge-factored, non-projective spanning-tree models. Our algorithms exploit a fundamental connection between gradients and expectations, which allows us to derive efficient algorithms. These algorithms are easy to implement, given the prevalence of automatic differentiation software. We motivate the development of our framework with several cautionary tales of previous (0) {re-search => research} , which has developed numerous less-than-optimal algorithms for computing expectations and their gradients. We demonstrate how our framework efficiently computes several quantities with known algorithms, including the expected attachment score, entropy, and generalized expectation criteria. As a bonus, we give algorithms for quantities that are missing in the literature, including the KL divergence. In all cases, our approach matches the efficiency of existing algorithms and, in several cases, (1) {reducesthe => reduces the} runtime complexity by a factor (or two) of the sentence length. We validate the implementation of our framework through runtime experiments. We find our algorithms are (2) {upto => up to} 12 and 26 times faster than previous algorithms for computing the Shannon entropy and the gradient of the generalized expectation objective, respectively.	We give a general framework for inference in spanning tree models. We propose unified algorithms for the important cases of first-order expectations and second-order expectations in edge-factored, non-projective spanning-tree models. Our algorithms exploit a fundamental connection between gradients and expectations, which allows us to derive efficient algorithms. These algorithms are easy to implement, given the prevalence of automatic differentiation software. We motivate the development of our framework with several cautionary tales of previous re-search, which has developed numerous less-than-optimal algorithms for computing expectations and their gradients. We demonstrate how our framework efficiently computes several quantities with known algorithms, including the expected attachment score, entropy, and generalized expectation criteria. As a bonus, we give algorithms for quantities that are missing in the literature, including the KL divergence. In all cases, our approach matches the efficiency of existing algorithms and, in several cases, (0) {reducesthe => reduces the} runtime complexity by a factor (or two) of the sentence length. We find our algorithms are (1) {upto => up to} 12 and 26 times faster than previous algorithms for computing the Shannon entropy and the gradient of the generalized expectation objective, respectively.	doc_rev_1

Figure 7: A screenshot of the iterative document-revision quality evaluation interface for human evaluators.

H Details on Automatic Evaluation for Model Revisions

Table 14 provides detailed automatic evaluation results for FELIX and PEGASUS, including SARI, BLEU, and ROUGE. We choose these automatic metrics following prior text revision works (Malmi et al., 2019; Dong et al., 2019; Mallinson et al., 2020). Note that the KEEP score of Baseline is not 100 because the source sentence keeps all n-grams, but there might be certain n-grams that are not kept in the reference sentence. This results in the non-perfect KEEP score since both recall and precision are calculated.

Table 15 further provides SARI score under different revision depths as well as different edit-intentions. We find that PEGASUS only conduct deletions in the revision depth 3, and the SARI score for each edit-intention varies a lot across different revision depths.

Table 16 and Table 17 are some examples of iterative text revisions generated by FELIX and PEGASUS trained on ITERATER-FULL. We observe that while FELIX can make more edits with more iterations than PEGASUS, it cannot ensure the quality of its generated edits. FELIX often insert some random out-of-context tokens into the original text, and distort the semantic meaning of the original text. PEGASUS is better at preserving the semantic meaning of the original text, but it is more likely to delete phrases or tokens in deeper revision depth.

Model	Training Data	SARI	DEL	ADD	KEEP	BLEU	R-1	R-2	R-L	Avg.
FELIX	ITERATER-HUMAN-RAW	29.23	19.23	0.62	67.85	49.48	77.27	60.11	63.43	47.38
FELIX	ITERATER-HUMAN	30.65	20.26	0.99	70.71	54.35	78.97	58.46	59.06	48.02
FELIX	ITERATER-FULL-RAW	30.34	20.44	1.40	69.18	55.10	76.47	58.07	56.49	47.31
FELIX	ITERATER-FULL	**33.48**	**22.39**	**2.52**	**75.52**	**61.90**	**80.65**	**64.97**	**63.72**	**53.03**
BART	ITERATER-HUMAN-RAW	33.20	9.81	3.58	86.20	**78.59**	85.93	79.94	85.20	65.66
BART	ITERATER-HUMAN	34.77	13.43	**5.91**	84.97	74.43	85.23	79.00	84.45	64.55
BART	ITERATER-FULL-RAW	33.88	12.38	2.34	**86.92**	78.55	86.66	**80.97**	86.05	66.16
BART	ITERATER-FULL	**37.28**	**19.83**	5.69	86.33	77.50	**86.85**	80.43	**86.14**	**66.97**
PEGASUS	ITERATER-HUMAN-RAW	33.09	10.61	1.57	87.09	**79.09**	87.50	81.65	86.77	66.32
PEGASUS	ITERATER-HUMAN	34.43	13.26	2.89	87.14	78.85	87.53	81.77	86.84	66.71
PEGASUS	ITERATER-FULL-RAW	34.67	13.93	2.36	**87.53**	78.21	**87.63**	**82.02**	**87.06**	66.65
PEGASUS	ITERATER-FULL	**37.11**	**19.66**	**4.44**	87.16	77.60	87.42	81.84	86.84	**67.18**
Baseline	-	29.47	0.0	0.0	88.42	81.25	88.67	83.51	88.04	66.25

Table 14: Model performances evaluated on the test set of ITERATER-HUMAN. R-1, R-2, and R-L refer to ROUGE-1, ROUGE-2, and ROUGE-L metrics, respectively, and Avg is computed by taking the mean of SARI, BLEU, and R-L scores. Baseline refers to a no-edit baseline, where we simply use the input text as the output.

t	Edit-Intention	SARI	ADD	DEL	KEEP
1	FLUENCY	46.22	18.53	24.00	96.12
1	COHERENCE	38.33	6.42	17.91	90.66
1	CLARITY	34.35	1.72	1.72	82.54
1	STYLE	40.61	0.0	32.63	89.19
2	FLUENCY	30.71	0.0	0.0	92.14
2	COHERENCE	29.50	0.0	6.25	82.26
2	CLARITY	35.29	7.33	18.94	86.19
2	STYLE	30.34	0.0	0.0	91.04
3	FLUENCY	32.74	0.0	3.98	94.26
3	COHERENCE	37.18	0.0	21.13	90.41
3	CLARITY	34.62	0.0	21.04	82.81
3	STYLE	32.09	0.0	37.50	58.77

Table 15: Detailed SARI scores for PEGASUS trained on ITERATER-FULL and evaluated on the test set of ITERATER-HUMAN, where t is the revision depth.

t	FELIX	PEGASUS	HUMAN
0	The three shareholders had unanimously agreed not to appoint an auditor for the company , but according to Investigate Magazine, another shareholder Russell Hyslop, had never been consulted about the matter.	The three shareholders had unanimously agreed not to appoint an auditor for the company , but according to Investigate Magazine, another shareholder Russell Hyslop, had never been consulted about the matter.	The three shareholders had unanimously agreed not to appoint an auditor for the company , but according to Investigate Magazine, another shareholder Russell Hyslop, had never been consulted about the matter.
1	~~T~~the three shareholders had unanimously agreed not to appoint an auditor for the company , but according to ~~I~~investigate ~~M~~magazine , ~~another shareholder Russell Hyslop~~, had never been consulted about the matter, another .	The three shareholders had unanimously agreed not to appoint an auditor for the company~~,~~. ~~but a~~According to Investigate Magazine, another shareholder Russell Hyslop, had never been consulted about the matter.	The three shareholders had unanimously agreed not to appoint an auditor for the company~~,~~; but according to Investigate Magazine, another shareholder Russell Hyslop, had never been consulted about the matter.
2	the three shareholders had unanimously agreed not to appoint an auditor for the company , but according to ~~investigate magazine~~, had never been consulted about the matter , another investigate .	The three shareholders had unanimously agreed not to appoint an auditor for the company. According to ~~Investigate Magazine~~, another shareholder Russell Hyslop, had never been consulted about the matter.	The three shareholders had unanimously agreed not to appoint an auditor for the company~~:~~, but according to Investigate Magazine, another shareholder, Russell Hyslop, had never been consulted about the matter.
3	the three shareholders had unanimously agreed not to appoint an auditor for the company , had never been consulted about the matter , but according to reuters , no . g launched ~~had never been consulted about the matter ,~~ another investigate .	The three shareholders had unanimously agreed not to appoint an external auditor for the company. According to, another shareholder Russell Hyslop, had never been consulted about the matter.	-
4	the three shareholders had unanimously agreed not to appoint an auditor for the company , had never been consulted about the matter , but according to ~~reuters~~, no . g launched another investigate .	The three shareholders had unanimously agreed not to appoint an external auditor for the company. According to~~,~~ another shareholder Russell Hyslop, had never been consulted about the matter.	-
5	the three shareholders had unanimously agreed not to appoint an auditor for the company , had never been consulted about the matter , but according to ~~, no . g~~ cre et company . the company launched another investigate ~~:~~inc . , no . g	-	-
6	the three shareholders had unanimously agreed not to appoint an auditor for the company , had never been consulted about the matter , but ~~according to cre et company . the company~~ launched another investigate inc . , no . g .	-	-

Table 16: A sample snippet of iterative text revisions generated by FELIX and PEGASUS (both trained on ITERATER-FULL, conditioned on Fluency intention) respectively, where $t = 0$ indicates the original input text.

t	FELIX	PEGASUS	HUMAN
0	Trainee programs and graduate programs are arranged by private companies and public sector employers where the trainee is offered the possibility to take part 6 to 20 months training programs . During the duration of these programs, the trainee is expected to receive a salary as well as is expected to have full-time employment awaiting in the company when the program is over.	Trainee programs and graduate programs are arranged by private companies and public sector employers where the trainee is offered the possibility to take part 6 to 20 months training programs . During the duration of these programs, the trainee is expected to receive a salary as well as is expected to have full-time employment awaiting in the company when the program is over.	Trainee programs and graduate programs are arranged by private companies and public sector employers where the trainee is offered the possibility to take part 6 to 20 months training programs . During the duration of these programs, the trainee is expected to receive a salary as well as is expected to have full-time employment awaiting in the company when the program is over.
1	~~T~~jobs , training and graduate programs are arranged by private companies and public sector employ~~ers where the trainee is offered the possibility to take part 6 to 20 months training programs . During the duration of these programs, the trainee is expected to receive a salary as well as is expected to have full-time employment awaiting in the company~~ment awaiting in full - time salary as well as is expected to receive a duration of these programs , the company when the program is over . employers where the trainee programs is expected to have part time . during training the trainee as the trainees assigned as the	Trainee programs and graduate programs are arranged by private companies and public sector employers where the trainee is offered the possibility to take part ~~6~~six to 20 months training programs. During the duration of these programs, the trainee is expected to receive a salary as well as is expected to have full-time employment awaiting in the company when the program is over.	Trainee programs ~~and graduate programs~~ are arranged by private companies and public sector employers where the trainee is offered the possibility to take part 6 to 20 months training programs . During the duration of these programs, the trainee is expected to receive a salary as well as is expected to have full-time employment awaiting in the company when the program is over.
2	jobs, training and graduate programs are arranged by private companies and public sector employment awaiting in full - time salary as well as is expected to receive a duration of these programs , the company when ~~the program~~ is ~~over~~. ~~employ-ers where the trainee programs is expected tohave part time . during training the trainee as thetrainees assigned~~. as the	Trainee programs and graduate programs are arranged by private companies and public sector employers, where the trainee is offered the possibility to take part six to 20 months training programs. During the duration of these programs, the trainee is expected to receive a salary as well as is expected to have full-time employment awaiting in the company when the program is over.	Trainee programs and graduate programs are arranged by private companies and public sector employers where the trainee ~~is offered the possibility to take part 6 to 20 months training programs~~ position has a varied duration depending on the company's program . During the duration of these programs, the trainee is expected to receive a salary as well as is expected to have full-time employment awaiting in the company when the program is over.
3	jobs , training and graduate programs are arranged by private companies and public sector employment awaiting ~~in full - time salary as well as is expected to receive~~ a duration of these programs ~~the company when . as the~~, full - time salary expected to receive the company when received as well	Training programs and graduate programs are organized by private companies and public sector employers, where the trainee ~~is offered the possibility to take part six to 20 months training programs. During the duration of these programs, the trainee~~ is expected to receive a salary as well as is expected to have full-time employment awaiting in the company when the program is over.	-
4	jobs , training and graduate programs are arranged by private companies and public sector employment awaiting a duration of these programs , full - time salary expected to receive the company when received ~~as well~~	Training programs and graduate programs are organized by private companies and public sector employers, where the trainee is expected to receive a salary and ~~as well as~~ is expected to have full-time employment awaiting in the company when the program is over.	-

Table 17: A sample snippet of iterative text revisions generated by FELIX and PEGASUS (both trained on ITERATER-FULL, conditioned on Clarity intention) respectively, where $t = 0$ indicates the original input text.