ARXIVEDITS: Understanding the Human Revision Process in Scientific Writing

Chao Jiang1, Wei Xu1, Samuel Stevens2*

1 School of Interactive Computing, Georgia Institute of Technology
2 Department of Computer Science and Engineering, Ohio State University
chaojiang@gatech.edu wei.xu@cc.gatech.edu stevens.994@osu.edu

Abstract

Scientific publications are the primary means to communicate research discoveries, where the writing quality is of crucial importance. However, prior work studying the human editing process in this domain mainly focused on the abstract or introduction sections, resulting in an incomplete picture. In this work, we provide a complete computational framework for studying text revision in scientific writing. We first introduce ARXIVEDITS, a new annotated corpus of 751 full papers from arXiv with gold sentence alignment across their multiple versions of revision, as well as fine-grained span-level edits and their underlying intentions for 1,000 sentence pairs. It supports our data-driven analysis to unveil the common strategies practiced by researchers for revising their papers. To scale up the analysis, we also develop automatic methods to extract revision at document-, sentence-, and word-levels. A neural CRF sentence alignment model trained on our corpus achieves 93.8 F1, enabling the reliable matching of sentences between different versions. We formulate the edit extraction task as a span alignment problem, and our proposed method extracts more fine-grained and explainable edits, compared to the commonly used diff algorithm. An intention classifier trained on our dataset achieves 78.9 F1 on the fine-grained intent classification task. Our data and system are released at tiny.one/arxivedits.

1 Introduction

Writing is essential for sharing scientific findings. Researchers devote a huge amount of effort to revising their papers by improving the writing quality or updating new discoveries. Valuable knowledge is encoded in this revision process. Up to January 1st, 2022, arXiv (https://arxiv.org), an open access e-print service, has archived over 1.9 million papers, among which more than 600k papers have multiple versions available. This provides an amazing data source for studying text revision in scientific writing. Specifically, revisions between different versions of papers contain valuable information about logical and structural improvements at document-level, as well as stylistic and grammatical refinements at sentence- and word-levels. It also can support various natural language processing (NLP) applications, including writing quality assessment and error correction (Louis and Nenkova, 2013; Xue and Hwa, 2014; Daudaravicius et al., 2016; Bryant et al., 2019), text simplification and compression (Xu et al., 2015; Filippova et al., 2015), style transfer (Xu et al., 2012; Krishna et al., 2020), hedge detection (Medlock and Briscoe, 2007), and paraphrase generation (Dou et al., 2022).

In this paper, we present a complete solution for studying the human revision process in the scientific writing domain, including annotated data, analysis, and system. We first construct ARXIVEDITS, which consists of 751 full arXiv papers with gold sentence alignment across their multiple versions of revisions, as shown in Figure 1. Our corpus spans 6 research areas, including physics, mathematics, computer science, quantitative biology, quantitative finance, and statistics, published in 23 years (from 1996 to 2019). To the best of our knowledge, this is the first text revision corpus that covers full multi-page research papers. To study sentence-level revision, we manually annotated fine-grained edits and their underlying intentions that reflect why the edits are being made for 1,000 sentence pairs, based on a taxonomy that we developed consisting of 7 categories.

Our dataset addresses two major limitations in prior work. First, previous researchers mainly focus on the abstract (Gábor et al., 2018; Kang et al., 2018; Du et al., 2022) and introduction (Tan and Lee, 2014; Mita et al., 2022) sections, limiting the generalizability of their conclusions. In addition, a sentence-level revision may consist of multiple fine-grained edits made for different purposes (see
an example in Figure 1). Whereas previous work either concentrates on the change of a single word or phrase (Faruqui et al., 2018; Piryazan et al., 2020) or extracts edits using the diff algorithm (Myers, 1986), which is based on minimizing the edit distance regardless of semantic meaning. As a result, the extracted edits are coarse-grained, and the intentions annotated on top of them can be ambiguous.

Enabled by our high-quality annotated corpus, we perform a series of data-driven studies to answer: what common strategies are used by authors to improve the writing of their papers? We also provide a pipeline system with 3 modules to automatically extract and analyze revisions at all levels. (1) A neural sentence alignment model trained on our data achieves 93.8 F1. It can be reliably used to extract parallel corpus for text-to-text generation tasks. (2) Within a revised sentence pair, the edit extraction is formulated as a span alignment task, and our method can extract more fine-grained and explainable edits compared to the diff algorithm. (3) An intention classifier trained on our corpus achieves 78.9 F1 on the fine-grained classification task, enabling us to scale up the analysis by automatically extracting and classifying span-level edits from the unlabeled revision data. We hope our work will inspire other researchers to further study the task of text revision in academic writing.

2 Constructing arXivEDITS Corpus

In this section, we present the detailed procedure for constructing the arXivEDITS corpus. After posting preprints on arXiv, researchers can automatically update the submission, and that constitutes the revisions. More specifically, a revision denotes two adjacent versions of the same paper.1 An article group refers to all versions of a paper on arXiv (e.g., v1, v2, v3, v4). In this work, we refer to the changes applied to tokens or phrases within one sentence as sentence-level revision. The document-level revision refers to the change of an entire or several sentences, and the changes to the paragraphs can be derived from sentences. Table 1 presents the statistics of document-level revision in our corpus. After constructing this manually annotated corpus, we use it to train the 3 modules in our automatic system as detailed at §4.

2.1 Data Collection and Preprocessing

We first collect metadata for all 1.6 million papers posted on arXiv between March 1996 and December 2019. We then randomly select 1,000 article groups from the 600k papers that have more than one versions available. To extract plain text from the LaTeX source code of these papers, we improved the open-source OpenDetex2 package to better handle macros, user-defined commands, and additional LaTeX files imported by the input commands in the main file.3 We find this method is less error-prone for extracting plain text, compared to

---

1For example, the paper titled “Attention Is All You Need” ([https://arxiv.org/abs/1706.03762](https://arxiv.org/abs/1706.03762)) has five versions on arXiv submitted by the authors, constituting four revisions (v1-v2, v2-v3, v3-v4, v4-v5).

2https://github.com/pkubowicz/opendetex

3Our code is released at [https://tiny.one/arxivedits](https://tiny.one/arxivedits)
using other libraries such as Pandoc\textsuperscript{4} used in (Cohan et al., 2018; Roush and Balaji, 2020). Among the randomly selected 1,000 article groups, we obtained plain texts for 751 complete groups, with a total of 1,790 versions of papers, that came with the original LaTeX source code and contained text content that was understandable without an overwhelming number of math equations. A breakdown of the filtered groups is provided in Appendix A.

2.2 Paragraph and Sentence Alignment
Sentence alignment can capture all document-level revision operations, including the insertion, deletion, rephrasing, splitting, merging, and reordering of sentences and paragraphs (see Figure 1 for an example). Therefore, we propose the following 2-step annotation method to manually align sentences for papers in the 1,039 adjacent version pairs (e.g., v0-v1, v1-v2) from the 751 selected article groups, and the alignments between non-adjacent version pairs (e.g., v0-v2) then can be derived automatically.

1. Align paragraphs using a light-weighted alignment algorithm that we designed based on Jaccard similarity (Jaccard, 1912) (more details in Appendix B). It can cover 92.1\% of non-identical aligned sentence pairs, based on a pilot study on 18 article pairs. Aligning paragraphs first significantly reduces the number of sentence pairs that need to be annotated.

2. Collect annotation of sentence alignment for every possible pair of sentences in the aligned paragraphs using Figure-Eight\textsuperscript{5}, a crowdsourcing platform. We ask 5 annotators to classify each pair into one of the following categories: aligned, partially-aligned, or not-aligned. Annotators are required to spend at least 25 seconds on each question. The annotation instructions and interface can be found in Appendix D. We embed one hidden test question in every five questions, and the workers need to maintain an accuracy over 85\% on the test questions to continue working on the task.

We skip aligning 4.6\% sentences that contain too few words or too many special tokens. They are still retained in the dataset for completeness, and are marked with a special token. More details about the annotation process are in Appendix A and B. In total, we spent $3,776 to annotate 13,008 sentence pairs from 751 article groups, with a 526/75/150 split for train/dev/test sets in the experiments of automatic sentence alignment in §4. The inter-annotator agreement is 0.614 measured by Cohen’s kappa (Artstein and Poesio, 2008). To verify the crowd-sourcing annotation quality, an in-house annotator manually aligns sentences for 10 randomly sampled groups with 14 article pairs. If assuming the in-house annotation is gold, the majority vote of crowd-sourcing annotation achieves an F1 of 94.2 on these 10 paper groups.

2.3 Fine-grained Edits with Varied Intentions
Sentence-level revision involves the insertion, deletion, substitution, and reordering of words and phrases. Multiple edits may be tangled together in one sentence, while each edit is made for different purposes (see an example in Figure 1). Correctly detecting and classifying these edits is a challenging problem. We first introduce the formal definition of edits and our proposed intention taxonomy, followed by the annotation procedure.

Definition of Span-level Edits. A sentence-level revision $R$ consists of the original sentence $s$, target sentence $t$, and a series of fine-grained edits $e_i$. Each edit $e_i$ is defined as a tuple $(s_{a:b}, t_{c:d}, I)$, indicating span $[s_a, s_{a+1}, ..., s_b]$ in the original sentence is transformed into span $[t_c, t_{c+1}, ..., t_d]$ in the target sentence, with an intention label $I \in \mathcal{I}$ (defined in Table 2). The type of edit can be recognized by spans $s_{a:b}$ and $t_{c:d}$, where $s_{a:b} = [\text{NULL}]$ indicating insertion, $t_{c:d} = [\text{NULL}]$ for deletion, $s_{a:b} = t_{c:d}$ representing reordering, and $s_{a:b} \neq t_{c:d}$ for substitution.

Edit Intention Taxonomy. We propose a new taxonomy to comprehensively capture the intention of text revision in the scientific writing domain, as shown in Table 2. Each edit is classified into one of the following categories: Improve Language, Correct Grammar/Typo, Update Content, and Adjust Format. Since our goal is to improve the writing

<table>
<thead>
<tr>
<th>Operation at Document-level</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td># of sent. insertion (0-to-1)</td>
<td>25,229</td>
</tr>
<tr>
<td># of sent. deletion (1-to-0)</td>
<td>17,315</td>
</tr>
<tr>
<td># of sent. rephrasing (1-to-1)</td>
<td>17,755</td>
</tr>
<tr>
<td># of sent. splitting (1-to-n)</td>
<td>378</td>
</tr>
<tr>
<td># of sent. merging (n-to-1)</td>
<td>269</td>
</tr>
<tr>
<td># of sent. fusion (m-to-n)</td>
<td>142</td>
</tr>
<tr>
<td># of sent. copying (1-to-1)</td>
<td>95,110</td>
</tr>
</tbody>
</table>

Table 1: Statistics of document-level revision in our ARXIVEDIT5 corpus, based on manually annotated sentence alignment.
<table>
<thead>
<tr>
<th>Intention Label</th>
<th>Definition</th>
<th>Example</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improve Language</td>
<td>Minor adjustment to improve the accuracy or specificness of the description.</td>
<td>Further, we suggest a relativistic-invariant protocol for quantum information processing communication.</td>
<td>28.6%</td>
</tr>
<tr>
<td>More Accurate/Specific</td>
<td>Make the text sound more professional or coherent without altering the meaning.</td>
<td>...due to hydrodynamic interactions among cells</td>
<td>11.5%</td>
</tr>
<tr>
<td>Improve Style</td>
<td>Simplify complex concepts or delete redundant content to improve readability.</td>
<td>These include new transceiver architecture (TXRU array connected architecture)</td>
<td>8.7%</td>
</tr>
<tr>
<td>Simplify</td>
<td>Other language improvements that don’t fall into the above categories.</td>
<td>...due to changes in fuels used</td>
<td>7.6%</td>
</tr>
<tr>
<td>Other</td>
<td>Correct Grammar/Typo</td>
<td>Fix grammatical errors, correct typos, or smooth out grammar needed by other changes.</td>
<td>0.8%</td>
</tr>
<tr>
<td>Update Content</td>
<td>Update large amount of scientific content, add or delete major fact.</td>
<td>...characterized by long range hydrodynamic term and self-generated force due to actin remodeling.</td>
<td>25.4%</td>
</tr>
<tr>
<td>Adjust Format</td>
<td>Adjust table, figure, equation, reference, citation, and punctuation etc.</td>
<td>Similarly to what we did in Figure Fig. [REF], the statistical results obtained by means of</td>
<td>28.8%</td>
</tr>
</tbody>
</table>

Table 2: A taxonomy \((T)\) of edit intentions in scientific writing revisions. In each example, text with red background denotes the edit. Span with strike-through means the content got deleted, otherwise is inserted.

quality, we further break the Improve Language type into four fine-grained categories. During the design, we extensively consult prior literature in text revision (Faigley and Witte, 1981; Fitzgerald, 1987; Daxenberger, 2016), edit categorization (Bronner and Monz, 2012; Yang et al., 2017), and analysis in related areas such as Wikipedia (Daxenberger and Gurevych, 2013) and argumentative essays (Zhang et al., 2017). The taxonomy is improved for several rounds based on the feedback from four NLP researchers and two in-house annotators with linguistic background.

**Annotating Edits.** In pilot study, we found that directly annotating fine-grained edits is a tedious and complicated task for annotators, as it requires separating and matching edited spans across two sentences. To assist the annotators, we use monolingual word alignment (Lan et al., 2021), which can find the correspondences between words and phrases with a similar meaning in two sentences, as an intermediate step to reduce the cognitive load during annotation. We find that, compared to strict word-to-word matching, edits usually have larger granularity and may cross linguistic boundaries. For example, in Figure 1, “corresponding to” and “correspondence with” should be treated as a whole to be meaningful and labeled an intention. Therefore, the edits can be annotated by adjusting the boundaries of the span alignment. We propose the following 2-step method that leverages word alignment to assist the annotation of edits:

1. Collect word alignment annotation by asking in-house annotators to manually correct the automatic word alignment generated by the neural semi-CRF word alignment model (Lan et al., 2021). The aligner is trained on the MTRef dataset and achieves state-of-the-art performance on the monolingual word alignment task with 92.4 F1.

2. Annotate edits by having in-house annotators inspect and correct the fine-grained edits that are extracted from word alignment using simple heuristics. The heuristics are detailed in §4.1. Two principles are followed during the correction: (1) Each edit should have a clear intention and relatively clear phrase boundaries; (2) Span pairs in substitution should be semantically related, otherwise should be treated as separated insertion and deletion.

We manually annotate insertion, deletion, substitution, and derive reordering automatically, since it can be reliably found by heuristics. Due to the slight variance in granularity, it is possible that more than one answer is acceptable. Therefore, we include all the alternative edits for sentence pairs in the dev and test sets in our annotation, among which 16% have more than one answer.

Overall, we found that our method can annotate more accurate and fine-grained edits compared to prior work that uses the diff algorithm. The diff method is based on minimizing the edit distance regardless of semantic meaning. Therefore, the extracted edits are coarse-grained and may contain many errors (detailed in Table 3).

**Annotating Intention.** As intentions can differ subtly, correctly identifying them is a challenging
task. Therefore, instead of crowdsourcing, we hire two experienced in-house annotators to annotate the intention for 2,122 edits in 1,000 sentence revisions. A two-hour training session is provided to both annotators, during which they are asked to annotate 100 sentence pairs and discuss until consensus. The inter-annotator agreement is 0.67 measured by Cohen Kappa (Artstein and Poesio, 2008), and 0.81 if collapsing the Improve Language category. The 1,000 sentence pairs are split into 600/200/200 for train/dev/test sets in experiments.

3 Analysis of Document-level Revisions

As a distinct style, scientific writing needs to be clear, succinct, precise, and logical. To understand what common strategies are used by authors to improve the writing of their papers, we present a data-driven study on document-level revisions in the scientific writing domain. This is enabled by our high-quality manually annotated corpus that consists of 1,790 versions of 751 full papers across 6 research areas in 23 years.

3.1 Distribution of Subjects and Versions

Figure 2 plots the statistics for the paper subjects and the number of versions. Physics (69.7%) and Math (14.8%) have the largest volume of multi-version papers, mainly due to the long history of use and a large number of sub-fields. About 26.7% papers have more than 2 versions available, enabling the study of iterative revisions. Figure 3 plots the length of the life cycle for each paper in our corpus, demonstrating a long-tail distribution.

3.2 Analysis of the Overall Update Ratio

We first investigate, in general, how much content is being updated for each paper during its life cycle, which can potentially affect the type of revisions contained therein. We define the Update Ratio as 1 minus the percentage of sentences being kept between two versions, which is derived from manually annotated sentence alignment.

Figure 4(a) presents how much content is being updated for each paper between its first and last versions. For papers that have two versions available, the distribution is heavily skewed towards the left end. The median update ratio is 19.0%, meaning that most papers have a mild revision. Whereas the distribution is much flatter for papers with multiple versions, indicating they are more likely to have a major revision in the life cycle. Interestingly, a peak appears at the tail of the distribution, which means 3.7% of the papers are almost completely rewritten. However, as shown in Figure 4(b), both types of papers have a similar distribution of update ratio for revisions between adjacent versions.
Research Areas. We hypothesize researchers in different areas may have different practices for revising their papers. Figure 4 visualizes the distribution of update ratio for papers on different subjects. Researchers in Statistics make more significant revisions to their papers compared to the CS area.

Time Interval. Intuitively, the time interval between submissions may correlate with the overall update ratio. We calculate the Pearson’s correlation between the update ratio and the time spent on the revision, which is measured by the difference in timestamps between adjacent submissions. The correlation values are 0.577 and 0.419 for papers that have two versions and multiple versions available, and both correlations are significant. Figure 6 visualizes the relationship. Researchers make quick submissions for small adjustments while spending more time on major revisions.

3.3 Analysis of the Updated Sentences

We explore the dynamic of document-level edit operations to answer: where will and how researchers update the sentences in their papers? The relative positions of the inserted, deleted, and revised sentences are visualized in Figure 7. Researchers, in general, revise more sentences at the beginning of a paper, while the insertion and deletion of sentences occur more in the latter parts. This makes sense because the abstract and introduction sections are usually frequently revised by the authors, since they are among the most important sections. As shown in Figure 8, revised sentences take the majority when update ratio is low. As more content is being modified, the insertion and deletion of sentences will become more dominant, which is likely to correspond to the major updates on the main body of papers.

3.4 Analysis of the Edit Intention

To understand why the researchers revised the sentences, we run our span-level edit extraction and intention classification system (details in §4) on all the revised sentences between adjacent versions in 751 article groups. The distribution of the intentions is visualized in Figure 9. Most of the language-related edits occur at the beginning of a paper. The aggregation is gradually reduced for grammar/typo- and content-related edits. The adjustments to format (punctuations, figures, tables, citations, etc.) span throughout the whole paper.

Figure 5: Update ratio for papers in different research areas. Papers in STAT have higher update ratios compared to papers in CS.

Figure 6: The relationship between update ratio and time between adjacent submissions.

Figure 7: The relative position of the sentences that are being inserted, deleted, and revised.

Figure 8: The composition of edit actions as the update ratio changes.

Figure 9: The distribution of intentions for span-level edits in the revised sentences in our corpus.
<table>
<thead>
<tr>
<th>Methods</th>
<th>Perf. on ≤5 edits</th>
<th>Perf. on All Revisions</th>
<th>% of Edit Types</th>
<th>Len. of Edits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P R F1 EM</td>
<td>P R F1 EM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semi-CRF Aligner$_{simple}$</td>
<td>89.8 90.1 90.0 85.9</td>
<td>87.5 87.7 87.6 80.5</td>
<td>32.9 26.7 40.4</td>
<td>4.66 4.98 2.21</td>
</tr>
<tr>
<td>Semi-CRF Aligner$_{parse}$</td>
<td>90.0 90.0 90.0 87.0</td>
<td>87.4 86.8 87.1 81.5</td>
<td>32.7 25.0 42.3</td>
<td>4.76 5.17 2.72</td>
</tr>
<tr>
<td>QA-align$_{simple}$</td>
<td>90.3 90.9 90.6 87.0</td>
<td>87.7 88.4 88.0 82.0</td>
<td>33.2 24.0 42.9</td>
<td>4.46 4.62 2.08</td>
</tr>
<tr>
<td>QA-align$_{parse}$</td>
<td>90.4 90.7 90.5 86.5</td>
<td>88.1 88.1 88.1 81.5</td>
<td>32.6 23.5 43.8</td>
<td>4.65 4.24 2.49</td>
</tr>
<tr>
<td>Latexdiff</td>
<td>79.9 78.6 79.3 75.7</td>
<td>76.2 74.3 75.3 70.0</td>
<td>26.2 14.4 59.3</td>
<td>3.89 4.27 4.73</td>
</tr>
</tbody>
</table>

Table 3: Performance of different edit extraction methods on the ARXIVEDITs testset. The **Len.** is measured by the number of tokens. We report performance on all sentence revisions, and on sentence pairs with ≤5 edits, which takes 92.5% of test data. The best and second best scores in each column are highlighted by **bold** and *underline*.

4 Automatic Edit Extraction and Intention Identification

As manual annotation is costly and time-consuming, we develop a pipeline system to automatically analyze the revisions at scale. Our system consists of sentence alignment, edit extraction, and intention classification modules, which are trained and evaluated on our annotated data. The methods and evaluation results of each step are detailed below. Example outputs from our system are presented in Figure 11.

4.1 Edits Extraction via Span Alignment

Prior work relies on diff algorithm to extract edits, which is based on string matching regardless of semantic meaning. To extract more fine-grained and explainable edits, we formulate the edit extraction as a span alignment problem. Given the original and revised sentences, the fine-grained edits are derived from span alignment using simple heuristics.

**Our Method.** We fine-tune two state-of-the-art word alignment models: neural semi-CRF model (Lan et al., 2021) and QA-Aligner (Nagata et al., 2020) on our ARXIVEDITs corpus, after training them on the MTRef dataset (Lan et al., 2021) first. Although sourced from the news domain, we find fine-tuning the models on MTRef, which is the largest monolingual word alignment corpus, helps to improve 4 points on the F1 score. When fine-tuning on ARXIVEDITs, the annotated edits are used as training labels, where substitutions are formulated as span alignment, insertions and deletions are the unaligned tokens, and the rest words will be aligned to their identical counterparts.

When running inference, the output edits are derived from span alignment using simple heuristics, where the insertions and deletions are unaligned tokens in the revised and original sentences, respectively. Substitutions are the non-identical span alignments. A simple post-processing step is applied to strip the identical words at the beginning and end of the substituted span pairs.

To enable more flexible granularity, we also design slightly more complex heuristics to extract edits by leveraging compositional span alignment. As shown in Figure 10, for each aligned word in two sentences, we iteratively traverse their parent nodes in two constituency parsing trees (Joshi et al., 2018) for max-level times to find the lowest ancestors in two trees that can resolve all the involved word alignment without conflict. Instead of separated word-to-word replacements, the two spans will be treated as a whole in the substitution. The max-level is a hyperparameter and can be adjusted to control the granularity of the extracted edits.

**Baseline** Diff (Myers, 1986) algorithm has been widely used in prior work to extract edits from text revision (Yang et al., 2017; Du et al., 2022). It is an unsupervised method based on dynamic programming for finding the longest common subsequences between two strings. Insertions and deletions are derived from unmatched tokens. Substitutions are derived from adjacent insertion and deletion pairs. Diff algorithm has many implementations with different heuristics for post-processing. We compare against its implementation in the latexdiff package, which is used in a recent work (Du et al., 2022).

Figure 10: Illustration for extracting edits by leveraging constituency parsing tree. In this example, two full spans that only have loose correspondence can be aligned.
**Results**  We report precision, recall, F1, and exact match (EM) for edit extraction. Table 2 presents the results on ARXIVEDITs testset. We report the performance on all 200 sentence pairs in the test set, and on a subset of sentence pairs with ≤ 5 edits, which take 92.5% of the test data and are more common in real applications. Using simple heuristics, both models finetuned on our dataset outperform the baseline method by more than 10 points in F1 and EM. In addition, enabling compositional span alignment by leveraging the constituency parsing tree can increase the granularity of the extracted edits, as shown in the “Len. of Edits” column. For the latexdiff method, about 59.3% of extracted edits are span substitutions, with an average length of 4.73 tokens. This is because the diff method derives edits by minimizing the edit distance. Combining with the post-processing heuristics, latexdiff treats everything as large chunk substitutions regardless of their semantic similarity.

### 4.2 Intention Classification

Given an edit and the original/revised sentences, the goal here is to classify its edit intention. We formulate it in a way that is similar to the relation extraction task. We experiment with two competitive models: T5 (Raffel et al., 2020) and PURE (Zhong and Chen, 2021). The input is the concatenation of two sentences, where the edited spans are surrounded by special markers with the type (ins./del./subst.). The PURE model predicts the intention by classification, and the T5 model will generate the intention string.

**Results.**  Table 4 shows the results for both fine-grained and coarse-grained (collapsing the Improve Language category) classification experiments. Collapsing labels helps to improve the performance in the 4-way classification task, where a T5-large model achieves an accuracy of 84.4. Though it’s challenging to pick up the differences between 7 types of intentions, the T5-large model trained with fine-grained labels achieves an accuracy of 79.3. The per-category performance of the best-performing T5 model is presented in Table 5. It performs well in separating top-layer categories. Within the Improve Language type, it also achieves reasonable performance on Accurate and Simplify categories, while fall short on Style, which is likely due to the inherited difficulty in identifying language style.

### 4.3 Sentence Alignment

Accurate sentence alignment is crucial for reliably tracking document-level revisions. Prior work mainly relies on surface-level similarity metrics, such as BLEU score (Faruqui et al., 2018; Fallings et al., 2021) or Jaccard coefficients (Xu et al., 2015), combined with greedy or dynamic programming algorithms to match sentences. Instead, we finetune a supervised neural CRF alignment model on our annotated dataset. The neural CRF aligner is shown to achieve better performance at aligning sentences from articles with different readability levels in the Newsela Corpus (Jiang et al., 2020).

**Our Methods.**  We first align paragraphs using the light-weighted paragraph alignment algorithm we designed (more details in Appendix B). Then, for each aligned paragraph pair, we apply our trained neural CRF alignment model to align sentences from both the old to new version and the reversed directions. The outputs from both directions are merged by intersection.

**Results**  We report precision, recall, and F1 on the binary classification task of aligned + partially-aligned vs. not-aligned. Table 6 presents the experimental results on ARXIVEDITs testset. It is worth noticing that the identical sentence pairs are excluded during the evaluation as they are trivial

<table>
<thead>
<tr>
<th>Intention Label</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjust Format</td>
<td>96.7</td>
<td>94.6</td>
<td>95.6</td>
</tr>
<tr>
<td>Update Content</td>
<td>84.8</td>
<td>86.9</td>
<td>85.8</td>
</tr>
<tr>
<td>Fix Grammar/Typo</td>
<td>81.1</td>
<td>85.1</td>
<td>83.1</td>
</tr>
<tr>
<td>Language-Simplify</td>
<td>75.0</td>
<td>66.7</td>
<td>70.6</td>
</tr>
<tr>
<td>Language-Accurate</td>
<td>54.7</td>
<td>63.0</td>
<td>58.6</td>
</tr>
<tr>
<td>Language-Style</td>
<td>46.9</td>
<td>37.5</td>
<td>41.7</td>
</tr>
</tbody>
</table>

Table 5: Breakdown performance of the best performing T5-large model on ARXIVEDITs testset for fine-grained intention classification task.

![Table 4](image-url)
If furthermore the [MATH] body Efimov states remain after extension of the Hilbert space to allow all correlations, we can expect these sequences to be continued to the threshold.

Update Content

Figure 11: Two example outputs from our Semi-CRF Aligner system. The intentions are predicted by our best-performing T5 model.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Char. 3-gram (Štajner et al.)</td>
<td>87.7</td>
<td>87.7</td>
<td>87.7</td>
</tr>
<tr>
<td>TF-IDF (Paetzold et al.)</td>
<td>90.3</td>
<td>91.6</td>
<td>90.9</td>
</tr>
<tr>
<td>Jaccard (Xu et al.)</td>
<td>90.7</td>
<td>89.5</td>
<td>90.1</td>
</tr>
<tr>
<td>BLEU (Faruqui et al.)</td>
<td>89.9</td>
<td>89.6</td>
<td>89.8</td>
</tr>
<tr>
<td>Neural CRF Aligner$\text{Dual}$ (Ours)</td>
<td><strong>96.9</strong></td>
<td><strong>91.0</strong></td>
<td><strong>93.8</strong></td>
</tr>
</tbody>
</table>

Table 6: Evaluation Results of different sentence alignment methods on our ARXIVEDITS testset.

to classify and will inflate the performance. For the similarity-based method, we tune a threshold based on the maximal F1 on the devset. By training the state-of-the-art neural CRF sentence aligner on our dataset and merging the output from both directions, we are able to achieve 93.8 F1, outperforming other methods by a large margin. It is worth noticing that the precision of our model is particularly high, indicating that it can be reliably used to extract high-quality aligned sentence pairs, which can be used as the training corpus for downstream text-to-text generation tasks.

5 Related Work

Text Revision in Scientific Writing. As a clear and concise style of writing, various aspects of scientific writing has been studied in previous work, including style (Bergsma et al., 2012), quality (Louis and Nenkova, 2013), hedge (Medlock and Briscoe, 2007), paraphrase (Dong et al., 2021), statement strength (Tan and Lee, 2014), and grammar error correction (Daudaravicius et al., 2016). Prior work studying scientific writing mainly focuses on the abstract and introduction sections (Tan and Lee, 2014; Du et al., 2022; Mita et al., 2022). In comparison, we develop methods to annotate and automatically analyze full research papers. Our work mainly focuses on the writing quality aspect.

Edit and Edit Intention. Previous work in studying the human editing process (Faruqui et al., 2018; Pryzant et al., 2020) mainly focuses on the change of a single word or phrase, as it is hard to pair complex edits in both sentences. Our work is able to extract more fine-grained and interpretable edits by leveraging span alignment. Several prior work utilizes the intention to categorize edits and as a clue to understanding the purpose of the revision. Some of their intention taxonomies focus on a specific domain, such as Wikipedia (Yang et al., 2017; Anthonio et al., 2020) and argumentative essay (Zhang et al., 2017; Kashefi et al., 2022). The intention taxonomy in our work is built on top of prior literature, with an adaptation to the scientific writing domain.

6 Conclusion

In this paper, we present a comprehensive study that investigates the human revision process in the scientific writing domain. We first introduce ARXIVEDITS, a new annotated corpus of 751 full arXiv papers with gold sentence alignment across their multiple versions of revisions, and fine-grained span-level edits together with their underlying intents for 1,000 sentence pairs. Based on this high-quality annotated corpus, we perform a series of data-driven studies to analyze the common strategies used by the researchers to improve the writing of their papers. In addition, we develop automatic methods to analyze revision at document-, sentence-, and word-levels. Our annotated dataset, analysis, and automatic system together provide a complete solution for studying text revision in the scientific writing domain.
Limitations

Due to the user groups of arXiv, our corpus mainly covers research papers in the field of science and engineering, while doesn’t contain articles from other areas, such as philosophy and arts. In addition, future work could investigate research papers that are written in non-English languages.

Acknowledgement

We thank three anonymous reviewers for their helpful feedback on this work. We also thank Andrew Duffy and Manish Jois for their help with annotations and human evaluation. This research is supported in part by the NSF awards IIS-2144493 and IIS-2055699. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of NSF or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References


A Details of Preprocessing

We randomly sample 1,000 paper IDs from arXiv that have multiple versions available and download their LaTeX source code for all the versions. During the pre-processing process, we aim to keep each article group complete. About 105 versions of papers don’t have source code available. After removing them, 959 groups are complete. There are 4 groups of papers using the harvmac package, which will introduce detex errors; after removing them, 955 groups are left. We then remove 162 groups of math-heavy papers and 42 groups of extremely short papers. After the filtering process, 751 complete groups of papers are left.

Among 335k sentences from all versions in the 751 article groups, we skip aligning 4.6% sentences that contain too few words or too many special tokens. Most of the 4.6% skipped content is (a) unusually short (\(<3\) tokens) and math-heavy text (\(\geq60\%\) special tokens) in math papers or (b) occasional de-tex errors (\(>1000\) char). They don’t contain much natural language content that can be analyzed or leveraged. In addition, annotators reported that such text increases the difficulty of aligning sentences. The criteria are detailed below.

We will skip aligning a paragraph if it meets one of the following criteria:

• Contains < 10 tokens.
• Contains > 30% special tokens (inline/block math, citation, and references).

We will skip aligning a sentence if it meets one of the following criteria:

• Contains > than 1,000 characters.
• Contains \(\leq 3\) tokens.
• Contains > 60% special tokens (inline/block math, citation, and references).
• Contains \(\geq 70\%\) English characters.
• Ends with "," or "":".

We also detect citations, references, inline math symbols, block equations, and present them as special markers which are easier to read for the annotators.

B Details of Paragraph Alignment and the Annotation Process

We design a light-weighted automatic paragraph alignment algorithm based on Jaccard similarity, which can cover 92.1% of non-identical sentence alignment in the pilot study. The algorithm is shown in Algorithm 1. The lengths of the two documents are represented by \(k\) and \(l\). \(d\) denotes the difference of relative position for two paragraphs with indices of \(i\) and \(j\), where \(d(i, j) = |\frac{r_i}{k} - \frac{r_j}{l}|\). The hyperparameters \(\tau_1 = 0.28, \tau_2 = 0.15, \tau_3 = 0.85, \tau_4 = 0.2\) are tuned on the devset.

At a high level, given an article pair, we first calculate the pairwise similarities for all possible paragraph pairs using the first block of the algorithm. Paragraph pairs are aligned by the second and third blocks of the algorithm if their similarity and relative distance reach certain thresholds.

To improve the money efficiency when annotating sentence alignment, we design a hybrid method to collect alignment labels for each sentence pair in the aligned paragraph pairs. We found that sentence pairs with Jaccard similarity > 0.7 and < 0.2 can be reliably automatically labeled as aligned and not-aligned. The thresholds are determined based on a pilot study and can achieve nearly 100% precision. Therefore, we automatically labeled the sentence pairs with Jaccard similarity > 0.7 and < 0.2, and collected human annotation for the rest of the candidate sentence pairs on the Figure Eight platform.

Algorithm 1: Paragraph Alignment Algorithm

```
Initialize: \(\text{alignP} \in \mathbb{R}^{k \times l} \) to 0
Initialize: \(\text{simP} \in \mathbb{R}^{2 \times k \times l} \) to 0
for \(i \leftarrow 1 \) to \(k\) do
    for \(j \leftarrow 1 \) to \(l\) do
        \(\text{simP}[1, i, j] = \frac{\max_{c_p \in C_p, c_q \in C_q} \text{simSent}(s_p, c_q)}{\sum_{c_p \in C_p} \text{simSent}(s_p, c_q)}\)
        \(\text{simP}[2, i, j] = \frac{\max_{c_p \in C_p} \text{simSent}(s_p, c_q)}{\sum_{c_q \in C_q} \text{simSent}(s_p, c_q)}\)
    end
end
for \(j \leftarrow 1 \) to \(l\) do
    \(i_{\text{max}} = \arg\max \text{simP}[2, i, j]\)
    if \(\text{simP}[1, i_{\text{max}}, j] > \tau_1 \) and \(d(i_{\text{max}}, j) < \tau_2\) then
        \(\text{alignP}[i_{\text{max}}, j] = 1\)
    else if \(\text{simP}[1, i_{\text{max}}, j] > \tau_3\) then
        \(\text{alignP}[i_{\text{max}}, j] = 1\)
    end
for \(i \leftarrow 1 \) to \(k\) do
    \(j_{\text{max}} = \arg\max \text{simP}[1, i, j]\)
    if \(\text{simP}[2, i, j_{\text{max}}] > \tau_1 \) and \(d(i, j_{\text{max}}) < \tau_4\) then
        \(\text{alignP}[i, j_{\text{max}}] = 1\)
    else if \(\text{simP}[2, i, j_{\text{max}}] > \tau_3\) then
        \(\text{alignP}[i, j_{\text{max}}] = 1\)
    end
end
return \(\text{alignP}\)
```
C Experiment Details

Our experiments are run on $4 \times A40$ GPUs. The implementation and hyperparameter tuning process are detailed below, where the one marked with $*$ performs best. We perform 3 runs for each setting, and average the performance. We use scikit-learn package to calculate the precision, recall and F1.6

**Sentence Alignment.** We use the author’s implementation of the neural CRF sentence alignment model and initialize it with the pretrained SciBERT-based-uncased encoder (Beltagy et al., 2019). We tune the learning rate in $\{1e^{-5}, 3e^{-5}, 5e^{-5}\}$ based on F1 on the devset. The model is trained within 1.5 hours.

**Intention Classification.** We use the Huggingface7 implementation of the T5 model, and use the author’s implementation of the PURE model. We initialized the PURE model with SciBERT-based-cased encoder (Beltagy et al., 2019). We tune the learning rate in $\{1e^{-5}, 3e^{-5}, 5e^{-5}, 7e^{-5}\}$ based on F1 on the devset. Both models are trained within 1 hour.

**Edits Extraction.** We use the original author’s implementations for the neural semi-CRF word alignment model and the QA-Align model. We initialize the semi-CRF model with SpanBERT-large encoder (Joshi et al., 2020) and initialize the QA-Align model with SciBERT-based-uncased encoder (Beltagy et al., 2019). We use the default hyperparameters for both models. The semi-CRF model takes about 10 hours to train, and the QA-Align model takes about 3 hours to train.

---

7https://huggingface.co/
D Crowdsourcing Annotation Interface

D.1 Screenshot of the Instructions

Instructions

- A and B are equivalent
  - Case 1: A simplify B or B simplify A (equivalent in meaning, though differ in length):

A: The results of stationary transport using the nonequilibrium Green's function (NEGF) formalism have been reported in the literature [CITATION].
B: The results of stationary transport for nonequilibrium systems have been reported by many authors [CITATION].

Two sentences convey the same meaning, while one sentence is simpler than the other one.

- Case 2: A and B are equivalent in both meaning and readability:

A: In Figure [REF] (b) a sample of trajectories illustrating the dynamics associated with the results of Figure [REF] (a) is displayed.
B: A sample of reduced trajectories illustrating the dynamics associated with the results of Fig. [REF](a) is displayed in part (b) of the same figure.

Two sentences are completely equivalent, as they mean the same thing.

- A and B are partially overlapped:
  - Case 1:

A: The dynamics described by this equation leads to the correct intensity pattern when the statistics of a large particle ensemble is considered [CITATION] (see below in Section [REF]).
B: Due to the continuity equation [REF] and definition [REF], the dynamics described by Eq. [REF] leads to the correct intensity pattern when the statistics of a large particle ensemble is considered [CITATION](as also happens in standard Bohmian mechanics).

One sentence contains most of the information of the other one. It also contains important extra information.

- Case 2:

A: We calculate logical error rates of amplified error rates [MATH] that are near to the threshold and fit logical error rates with the function [EQUATION].
B: We calculate logical error rates corresponding to amplified error rates [MATH] and small size surface code lattices using the Monte Carlo method.

Two sentences share some information in common. And each of them also contains extra information.

- A and B are mismatched:

A: In the wire network shown in Fig. [REF](b), all PPs are performed with PP TUQSs of qubits and neighbouring stabilisers share measurement TUQSs.
B: The number of PP TUQSs can be reduced using the system shown in Fig. [REF](b), in which there is only one PP TUQS per qubit.

Two sentences are take about different issue.

Sometimes, the sentence pair may share some terms (like PP, TUQS in this example), but are not really equivalent or partially overlapped. You need to read them carefully and understand their meaning to make correct judgments.

Figure 12: Instructions for our crowdsourcing annotation of sentence alignments on the Figure Eight platform.
D.2 Screenshot of the Interface

Sentence A
The proposed method can be applied to find relations of the derived type giving possibility to search for other heavy virtual states.

Sentence B
The proposed method can be applied to find the relations of the derived type for different effective four-fermion operators generated by the [MATH] (for instance, discussed in Refs. [CITATION]).

What's the relationship between Sentence A and Sentence B?

- A and B are equivalent
  - Convey the same meaning, though one sentence can be much shorter or simpler than the other sentence.

- A, B are partially overlapped
  - A and B partially overlap (share information in common, while some important information differs/missing).

- A and B are mismatched
  - The two sentences are completely dissimilar in meaning.

Comments (Optional)
If you have any comment about this HIT, please type it here.

Figure 13: Interface for our crowdsourcing annotation of sentence alignments on the Figure Eight platform.