Re3: A Holistic Framework and Dataset for Modeling Collaborative Document Revision

Anonymous ACL submission

Abstract

Collaborative review and revision of textual documents is the core of knowledge work and a promising target for empirical analysis and NLP assistance. Yet, a holistic framework that would allow modeling complex relationships between document revisions, reviews and author responses is lacking. To address this gap, we introduce Re3, a framework for joint analysis of collaborative document revision. We instantiate this framework in the scholarly domain, and present Re3-Sci, a large corpus of aligned scientific paper revisions manually labeled according to their action and intent, and supplemented with the respective peer reviews and human-written edit summaries. We use the new data to provide first empirical insights into collaborative document revision in the academic domain, and to assess the capabilities of state-of-the-art LLMs at automating edit analysis and facilitating text-based collaboration. We make our annotation environment and protocols, the resulting data and experimental code publicly available.[1]

1 Introduction

Textual documents are a key medium of information exchange in the modern world. These documents often result from a collaboration of multiple individuals. The typical process of collaborative text production involves iterations of drafting, getting feedback (***re**views*), executing ***re**visions*, and providing ***re**sponses* that outline the implemented changes, serving as a vital element in facilitating effective communication (Cheng et al., 2020; Kuznetsov et al., 2022). Despite the importance of collaborative text revision and its high potential for NLP applications, we are missing a framework that formally describes this review-revision-response procedure grounded in real-world data. While prior work in NLP has studied relationships between original and revised documents (Du et al., 2022; Jiang et al., 2022), reviews and original documents (Dycke et al., 2023), reviews and revisions (Kuznetsov et al., 2022; D'Arcy et al., 2023), and reviews and responses (Gao et al., 2019; Cheng et al., 2020) – no prior frameworks allow jointly modeling all three components of text-based collaboration. Yet, such joint modeling is important as it provides deeper insights into the processes involved in text work, and opens new opportunities for NLP applications. Important tasks that involve reviews, revisions and responses such as *edit summarization* thus remain underexplored.

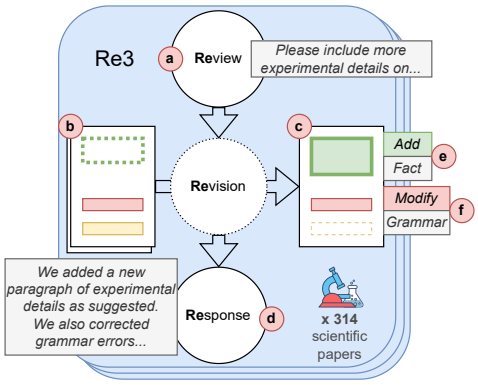


Figure 1: Re3 offers a holistic framework for studying the relationships between reviews (a), revisions (b-c) and responses (d) in text-based collaboration. It is instantiated in the Re3-Sci dataset that covers all edits in 314 full-length scientific publications manually labeled with edit action and intent (e) on different granularity levels, along with reviews that trigger edits and manually curated responses that summarize all edits made including self-initiated ones (f).

Comprehensive analysis of document-level revisions poses additional challenges. Contrary to sentence-level analysis, hierarchically structured documents (Ruan et al., 2022) bring distinct levels of granularity into editing. Individuals execute revisions at various granularity levels, with a range of actions and a spectrum of intents, reflecting the *what*, *how*, and *why* of the revisions (Figure 1 and §3.2). Realistic modeling of document revision

[1] URL omitted for anonymity

in text-based collaboration thus requires datasets and annotations that encompass the ***entire document context***, incorporating ***all edits*** made across various levels of ***granularity***, and providing qualitative labels for both ***action*** and ***intent***. We further term this kind of analysis as **full-scope** modeling of document revision. Prior research in NLP has primarily studied sentence-level edits while neglecting the broader document context (Daxenberger and Gurevych, 2012; Yang et al., 2017), variations in granularity (Du et al., 2022; Kashefi et al., 2022), and the underlying intent behind the edits (Spangher et al., 2022; Jiang et al., 2022). There is thus a gap in both methodologies and datasets for creating and analyzing full-scope annotations of document revisions, limiting our grasp of the intricate nature of the editing process.

To close this gap and enable a comprehensive study of text-based collaboration in NLP, we introduce **Re3**: the first holistic framework for modeling review, revision and response in collaborative writing (§3). We instantiate our framework in the scholarly domain and create **Re3-Sci**, the first large-scale human-annotated dataset that comprises 11.6k full-scope revision annotations for over 300 revised documents with substantial Inter-Annotator Agreement (IAA), as well as cross-document connections between reviews, revisions and responses (§4). Our framework and dataset, for the first time, enable large-scale empirical investigation of collaborative document revision, including edit localization and clustering within documents, edit mechanisms and motivations inferred through action and intent labels, and the impact of review requests (§5). Manually analyzing the complex relationships between reviews, revisions and responses is costly, and constitutes a promising NLP automation target. Facilitated by our data, we present a first exploration of the capability of large language models (LLMs) to address novel revision assistance tasks, such as review request extraction, revision alignment, edit intent classification and document edit summarization (§6). Our work thus makes four key **contributions**:

- A holistic framework for studying document revisions and associated interactions in collaborative writing, including label taxonomy and robust annotation methodology;
- A high-quality large-scale dataset that instantiates the framework in the domain of academic writing and peer review;
- An in-depth analysis of human editing behavior in the scholarly domain;
- Extensive experiments in automation with LLMs on four NLP tasks: review request extraction, revision alignment, edit intent classification and document edit summarization.

Our work paves the path towards comprehensive study of NLP for text-based collaboration in the scholarly domain and beyond.

2 Related Work

	length	edits	full-scope	align	intent
IteraTeR (2022)	197	7*	no	4k*	4k*
ArgRewrite (2022)	582	19	no	3.2k	3.2k
arXivEdits (2022)	3,916	17	no	**13k**	1k
Re3-Sci (ours)	**5,033**	**37**	**yes**	**11.6k**	**11.6k**

(a) Comparison of human-annotated document revision datasets. Presented are document length (words), average sentence edits per document, presence of full-scope revision annotations, and data size, i.e., count of aligned and labeled sentence edits. * refers to subsentence edits as only such annotations are available.

	full-scope	review-revision	revision-response	review-response
F1000RD (2022)	no	yes	no	no
NLPeer (2023)	no	yes	no	no
ARIES (2023)	no	yes	no	no
Re3-Sci (ours)	**yes**	**yes**	**yes**	**yes**

(b) Comparison of review-revision-response datasets. Presented are presence of full-scope revision annotations, and interactions between the documents.

Table 1: Related datasets comparison.

Document revision datasets. Research on text revision originates in studies on Wikipedia (Daxenberger and Gurevych, 2012; Yang et al., 2017; Faruqui et al., 2018) and academic writing (Tan and Lee, 2014; Xue and Hwa, 2014), which offer partial sentence-based annotations, neglecting the document context. Recent works have expanded the analysis to news articles (Spangher et al., 2022), student essays (Zhang et al., 2016; Kashefi et al., 2022), and scientific papers (Du et al., 2022; Jiang et al., 2022). However, some focus mainly on revision alignment yet overlook the underlying intents (Spangher et al., 2022; Jiang et al., 2022). Others restrict to specific sections or short texts, limiting analysis to sentence level (Du et al., 2022; Zhang et al., 2016; Kashefi et al., 2022). In this work, we introduce Re3-Sci, the first large-scale corpus with full-scope annotations of edit alignments, actions, and intents across multiple granularity levels in the entire document (Table 1a).

Collaborative revision in peer review. Scholarly peer review is an essential example of collaborative text work in the academic domain. Open peer review provides an excellent opportunity to study the review-revision-response procedure. Prior works in NLP for peer review investigate argument mining-driven review analysis (Hua et al., 2019; Fromm et al., 2020) and the interplay between reviews and argumentative rebuttals (Gao et al., 2019; Cheng et al., 2020; Bao et al., 2021; Kennard et al., 2022), among others. Only a few studies and datasets investigate revision requests in peer reviews and their connection to the original texts (Dycke et al., 2023), or to the actual revisions (Kuznetsov et al., 2022; D'Arcy et al., 2023). However, these do not provide full-scope annotations with qualitative labels and neglect self-initiated revisions not prompted by reviewers. Our work is the first to cover the entire review-revision-response procedure with full-scope annotations in the context of scholarly publishing and peer review (Table 1b).

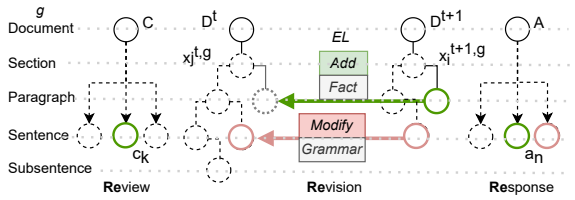


Figure 2: Illustration of the Re3 framework.

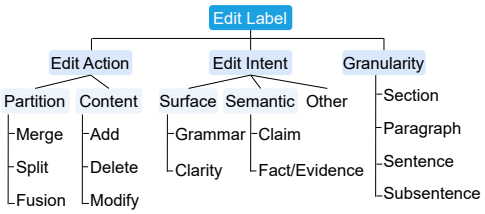


Figure 3: Re3 edit label taxonomy.

3 The Re3 Framework

The Re3 framework builds upon the recently introduced intertextual model by Kuznetsov et al. (2022). In particular, we represent documents as graphs that preserve document structure, allowing us to work on different levels of granularity, and treat cross-document relations as edges between the corresponding document graphs.

3.1 Model and Terminology

As shown in Figure 2, we conceptualize the review-revision-response procedure as a set of interactions among four document types - the original document D^t, the revised document D^{t+1}, the review C and the response A - along with the diverse types of connections between their text elements. Depending on the granularity g, text elements can vary from subsentence-level words and phrases to sentences, paragraphs, or sections. Text elements of granularity g of the old and new documents are noted as $x_j^{t,g} \in D^t$ and $x_i^{t+1,g} \in D^{t+1}$. Comparing two document versions, edit alignment links elements from the new and old versions. For analytical clarity, aligned elements maintain the same granularity in our study, noted as $e(x_i^{t+1,g}, x_j^{t,g})$. Each edit alignment e_{ij} is associated with an edit label, $EL_{ij} = (g, EA_{ij}, EI_{ij})$, which specifies the granularity, action, and intent of the edit, explaining *how* and *why* the edit is made to a text element of g (§3.2). When a new text element $x_i^{t+1,g}$ is added or an old one $x_j^{t,g}$ is deleted, the corresponding old or new element is null, noted as $e(x_i^{t+1,g}, null)$ or $e(null, x_j^{t,g})$.

Given that the reviews and responses are typically brief without rich structure, we focus on the sentence level in those documents. Reviews include requests $c_k \in C$ that may prompt edits. An addressed request c_k is linked to relevant edit e_{ij} as $ec(e_{ij}, c_k)$. A single request can lead to multiple edits, while self-initiated edits may not connect to any review request. Similarly, an author's response includes sentences $a_n \in A$ summarizing realized revisions. Each a_n connects to its respective edit e_{ij} via the relation $ea(e_{ij}, a_n)$.

3.2 Revision Dimensions and Label Taxonomy

We analyze revisions along three qualitative dimensions – *granularity*, *action*, and *intent* – and present our proposed label taxonomy in Figure 3. *Granularity* specifies the scope of the text elements subject to revision, which is crucial since the perception of revisions varies with granularity. For instance, extending a sentence may appear as adding text elements at subsentence level or as modifying an existing text element at sentence level (further exemplified in Table 12 in §A). In this work, we include section, paragraph, sentence, and subsentence granularities. *Action* specifies how revisions are made, including basic methods like addition, deletion, and modification, as well as complex operations like merges, splits, and fusions (elaborated in Table 13 in §A). *Intent* categorizes the underlying purpose into surface language improvements

for grammar or clarity and substantial semantic changes affecting claims or factual content, with detailed definitions and examples in Table 14 in §A. The three dimensions collectively characterize the nature, purpose, and significance of the revisions. For instance, factual content updates may entail sentence expansion with additional details or the incorporation of an entirely new sentence. When significant elaboration is necessary, new paragraphs or sections may be introduced. The taxonomy has been refined through feedback from two linguists and proved sufficient and crucial in the annotation study with six annotators (§4.3 and §B.6). The hierarchical structure of the taxonomy promotes easy expansion and adaptation across various domains by incorporating fine-grained labels.

4 Dataset Construction: Re3-Sci

4.1 Data Collection and Pre-processing

Scientific publishing, a prominent open source of collaborative document revision and review, offers ample data for our research objectives. We instantiate our framework based on the data from the F1000RD dataset (Kuznetsov et al., 2022) and the ARR-22 subset of the NLPeer corpus (Dycke et al., 2023), which include revisions of scientific papers along with their corresponding reviews, covering a range of fields including NLP, science policy, public health, and computational biology. Both datasets contain structured documents organized into section and paragraph levels, which we further refine to sentence level (§B.1). A total of 314 document pairs and related reviews are randomly selected for human annotation: 150 from NLPeer and 164 from F1000RD.

4.2 Pre-alignment

Identifying revision pairs from two lengthy documents is challenging, especially complicated by the expansive scope for comparison and the presence of recurring content. To address this, we employ a lightweight sentence alignment algorithm that systematically excludes identical pairs and identifies alignment candidates from the remaining sentences, considering both form and semantics similarity, as well as the document's context and structure (§B.2). Annotators are given the alignments and tasked with validating and correcting any alignment errors. Based on these corrections, the proposed algorithm achieves an accuracy of 0.95. The validated alignments are subsequently used for edit action and intent labeling.

4.3 Annotation Process

To perform the human annotation, we develop a cross-document annotation environment using INCEpTION (Klie et al., 2018), as detailed in §B.3. A pilot study with 20 document pairs is initiated to refine the label taxonomy, optimize the pre-alignment algorithm, improve the annotation tool, and develop comprehensive guidelines, with the assistance of three in-house annotators skilled in computer science or linguistics.

For annotation, six master's students with C1-level English proficiency are recruited (§B.5). We employ an iterative data quality management process to ensure the quality of the annotations. Initially, a 15-hour training session is conducted, involving a joint review of guidelines, live demonstrations, and practice annotations on a validation set of five document pairs. Given the initial suboptimal IAA in intent labeling, highlighting its complexity, we conduct further discussions on disagreements and common mistakes, followed by a final re-annotation of the validation set. This method ensures consistent comprehension of the guidelines and familiarity with the annotation process prior to actual annotation. Documents are divided into three data packages for iterative quality assessment, with intermediary meetings by the coordinator to address annotators' individual questions. The primary tasks, sentence-level revision alignment and edit labeling, are carried out by three annotators per sample. We release all three annotations with a gold label aggregated through majority voting. After annotation, we conducted an annotator survey to gather insights for future annotation studies.

We achieve a substantial (Landis and Koch, 1977) IAA of 0.78 Krippendorff's α for the labeling task and a perfect IAA of 1 Krippendorff's α for the alignment task. Table 5 in §B.4 shows progressive IAA improvement following iterative quality management between data packages, highlighting the method's effectiveness. As a qualitative assessment, the annotator survey (§B.6) confirms the adequacy of guidelines, label taxonomy, and annotation tool, as well as the effectiveness of iterative training. The annotators also highlight the effectiveness of the cross-document annotation environment, especially in aligning revision pairs, which potentially contributes to the perfect IAA in alignment. Further insights are provided in §B.6.

4.4 Statistics

	#doc	# S	# SS	# P	# Sec
Re3-Sci	314	11,648	3,038	5,417	2,008

(a) Count of aligned and labeled edits at sentence (S), subsentence (SS), paragraph (P), and section (Sec) levels.

	#review request	#review -revision	#revision -response
Re3-Sci	560	413	1,364

(b) Count of extracted review requests, their alignments with realized revisions, and linkages between revisions and edit summaries in response.

Table 2: Re3-Sci dataset statistics.

The Re3-Sci dataset comprises 314 document revision pairs. 11,648 sentence-level edits comprising sentence revision pairs, additions, and deletions are identified and annotated with respective edit action and intent labels. Based on the sentence-level annotations, 5,417 paragraph-level, and 2,008 section-level edits are identified. We also extract 3,038 subsentence-level edits from 2,008 sentence revision pairs, employing a constituency tree-based method similar to Jiang et al. (2022). These extractions and alignments are verified by a linguistic expert and labeled by three annotators. Furthermore, we randomly select 42 documents and extract 560 review sentences that may prompt changes. The review sentences are aligned with the corresponding revisions when possible, resulting in 413 linkages. Annotators summarize the document revisions in brief responses and then align a total of 784 summary sentences back to the related edits, resulting in 1,364 connections. See §B.5 for more details.

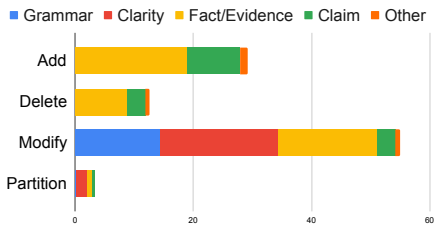
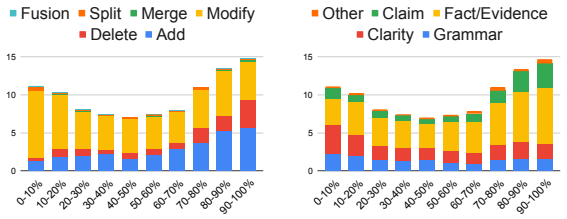


Figure 4: Distribution of sentence edit action and intent labels and their combinations in Re3-Sci.

5 Dataset Analysis

The framework and dataset allow us to answer new questions about human behavior in collaborative document revision in scholarly publishing.

RQ1: How and why do humans edit, and what are the relationships between edit actions and intents? Figure 4 reveals that authors predominantly modify existing content and add new material, with deletions being infrequent and partition changes even less common. It also suggests that the enhancement of fact or evidence is the primary focus of revisions, highlighting its importance in improving scientific quality. Moreover, Figure 4 illustrates that additions and deletions of sentences typically pertain to improving factual content or claims, but are never intended for superficial language enhancement. On the other hand, grammar and clarity improvements are usually realized by modifying existing sentences. This suggests that, from a modeling view, the edit action and intent labels may influence the prediction of each other.



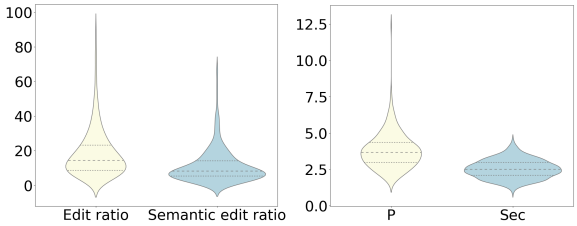
(a) Edit action distribution (b) Edit intent distribution

Figure 5: Edit action and intent distribution over the document. The x-axis represents the relative sentence positions within documents.

RQ2: How are edits distributed in documents? Figure 5 indicates that the initial and final parts of papers experience significantly more revisions. In terms of edit actions, the beginning of the document typically sees more modifications, while the end is characterized by a higher frequency of additions and deletions. Regarding edit intents, language enhancements for grammar or clarity are more common in the early parts, whereas changes affecting semantic content, such as facts or claims tend to occur more in the later parts. These suggest that the document position may be a valuable predictor for identifying edit actions and intents.

RQ3: How significant are the differences between document versions? To gauge the magnitude of change, we introduce the *Edit Ratio* metric, determined by the ratio of sentence edits to the sentence count in the original document. While the edit ratio reflects the extent of differences, the significance of document revisions is highlighted by the *Semantic Edit Ratio*, which is calculated by the ratio of semantic edits labeled with Fact/Evidence or Claim. The average document edit ratio stands at 18.45%. Figure 6a and Figure 10 in §C show

that the majority of documents experience moderate revisions with an edit ratio of 5-25%, while a small proportion has an edit ratio exceeding 50%, and only a few documents appear to have been extensively rewritten. The average semantic edit ratio stands at 11.18%, with most documents showing 0-20% of their content undergoing significant change. Notably, documents with a high edit ratio often do not correspond to a high semantic edit ratio, suggesting that documents with extensive revisions typically exhibit language quality issues.



(a) Document Update (b) CF

Figure 6: (a) Document update measured by edit ratio and semantic edit ratio. (b) CF measured at paragraph (P) and section (Sec) level.

RQ4: How are edits clustered by paragraphs and sections? We use Crest Factor (CF), a concept borrowed from signal processing (Parker, 2017), to assess the concentration of edits. Using a vector of sentence edit counts in each paragraph or section, CF quantifies the peak amplitude of this distribution. A CF value of 1 signifies an even distribution. The average paragraph CF is 3.79, indicating a substantial concentration of edits within a limited number of paragraphs. This trend of high edit concentration in a few paragraphs is further illustrated in Figure 6b and Figure 11 in §C. When examined at the section level, the average CF is 2.54, indicating a moderate tendency towards clustering.

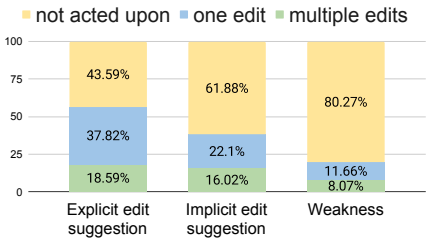


Figure 7: Three types of reviewers' requests and their impact. Displayed are proportions of requests not acted upon, actualized by single or multiple sentence edits.

RQ5: Are reviewers' requests acted upon? How are these realized in revision? Annotators categorize the relevant review requests into three types: *explicit edit suggestions* (27.9%), *implicit edit suggestions* (32.3%), and *general weakness comments* (29.8%). The first provides specific document locations and clear revision instructions; the second delivers guidance without locations; the last highlights general issues without specific suggestions. Figure 7 shows that more than half of explicit suggestions are implemented, with 18.59% actualized through multiple sentence edits. Implicit suggestions and general weakness comments are realized to a lesser extent. This implies that reviewers' explicit suggestions are more likely to be acted upon.

6 Automation with LLMs

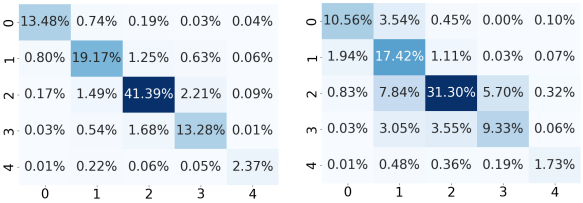
The Re3-Sci dataset facilitates a variety of NLP tasks, such as edit intent classification, document edit summarization, revision alignment and review request extraction. Here we discuss the automation of edit intent classification and document edit summarization with LLMs. We deem intent classification most interesting and relevant for NLP-assisted edit analysis, presenting challenges even for human annotators (§B.6). Document edit summarization emerges as a novel application designed to aid authors in crafting responses, which is, for the first time, enabled by our full-scope modeling approach, making a pioneering contribution to the field. Due to space limitations, we report additional experiments in revision alignment and review request extraction in §D.2 and §D.3. Computational details are provided in §D.4.

6.1 Edit Intent Classification

Task formulation and data. Formulated as a classification task given a sentence edit $e(x_i^{t+1,g}, x_j^{t,g})$, the objective is to predict the intent label EI_{ij}. For additions or deletions, only one sentence is used. We split the documents into 20% for training and 80% for testing.[2] The test set contains 5,045 revision pairs and 3,891 additions or deletions.

Models and methods. We evaluate Llama2-70B (Touvron et al., 2023) with multiple ICL demonstration selection methods and analyze CoT prompt formatting. The three dynamic ICL methods select the most similar demonstrations from the training set for each test sample using RoBERTa embeddings (Liu et al., 2019): *cat* uses cosine similarity of concatenated sentence embeddings, *diff* leverages the difference between sentence embeddings,

[2]We use training data for ICL example selection only and the rest for testing to get more reliable performance estimates.



(a) Human (b) Llama2-70B

Figure 8: Error analysis and human vs. LLM comparison in edit intent classification on the full test set. The y-axis presents the gold, the x-axis presents the annotated/predicted labels. The gold labels are the majority of the three human labels. The diagonal indicates the percentages of correct labels. 0: Grammar, 1: Clarity, 2: Fact/Evidence, 3: Claim, 4: Other.

and *loc* utilizes concatenated embeddings of the associated section titles. The static *def* method uses a default set of manually selected examples for each intent across all tests. For CoT, we instruct the model to predict the intent label (*L*) with rationale (*R*) in CoT style, evaluating how their order impacts results. A prompt example is provided in Table 11 in §D.1. Preliminary experiments indicate inadequate performance in jointly modeling revision pairs and single-sentence instances, leading us to separate experiments for each scenario.

Results and discussion. Table 3 shows the results on revision pairs. In addition to random and majority baselines, other baselines use the majority label of the top *n* selected training examples from the three proposed methods. Using the same examples for ICL, the *diff* method notably excels over others (block 1). Intriguingly, Llama2 doesn't merely depend on the majority of selected examples, as seen in comparison with the baselines - there is a significant improvement and the disparities between the methods diminish. Using five default examples outperforms *cat* and *loc*, and is on par with *diff* (block 2). Accuracy further increases when the gold label is accompanied by rationale in CoT style (i.e., L,R). However, reversing their order in both instruction and output (i.e., R,L) notably decreases performance. Combining default examples with rationale and dynamic examples further enhances accuracy (block 3). Altering the order of static and dynamic examples enhances results when using *diff*, though this is not consistent across all selection methods (blocks 3, 4). Omitting rationale from the default examples leads to a significant and consistent performance decline, highlighting the importance of CoT demonstrations (blocks 5, 6).

The best configuration involves three dynamic

Baselines					
Random	0.20	Majority	0.36		
diff1	0.45	cat1	0.38	loc1	0.31
diff3-maj	0.45	cat3-maj	0.38	loc3-maj	0.32
diff5-maj	0.46	cat5-maj	0.40	loc5-maj	0.34
diff8-maj	<u>0.47</u>	cat8-maj	0.41	loc8-maj	0.33
Our Models (ICL & CoT)					
①+ *dynamic examples*					
+diff1	0.60	+cat1	0.58	+loc1	0.56
+diff3	0.60	+cat3	0.57	+loc3	0.53
+diff5	<u>0.61</u>	+cat5	0.56	+loc5	0.52
+diff8	0.59	+cat8	0.56	+loc8	0.51
②+ *static examples*					
+def5	0.59	+def5 -(L,R)	<u>0.62</u>	+def5 -(R,L)	0.53
③+ *def5-(L,R) + dynamic*					
+diff1	0.62	+cat1	0.59	+loc1	0.59
+diff3	<u>0.63</u>	+cat3	0.59	+loc3	0.58
+diff5	<u>0.63</u>				
④+ *dynamic + def5-(L,R)*					
+diff1	0.64	+cat1	0.58	+loc1	0.60
+diff3	**<u>0.65</u>**	+cat3	0.59	+loc3	0.58
+diff5	0.63				
⑤+ *def5 + dynamic*					
+diff1	0.59	+cat1	0.58	+loc1	0.57
+diff3	<u>0.61</u>	+cat3	0.57	+loc3	0.55
+diff5	0.59				
⑥+ *dynamic + def5*					
+diff1	0.59	+cat1	0.57	+loc1	0.55
+diff3	0.61	+cat3	0.56	+loc3	0.54
+diff5	<u>0.62</u>				

Table 3: Llama2-70B accuracy in edit intents classification in revision pairs. Baselines are assessed on the full test set, subsequent models are evaluated on 20% of test samples for validation. Underlined is the best accuracy in the block, the highest accuracy is in bold.

diff-selected examples and the default examples with CoT rationale. This also yields the best performance for additions and deletions, as shown in Table 8 in §D.1. With this setup, joint evaluation on the full test set results in an accuracy of 0.7 and a macro-average F1 score of 0.69, significantly outperforming the baselines as shown in Table 10 in §D.1. The pronounced potential for advancement highlights the task's complexity for LLMs. This requires precise detection of changes and advanced reasoning capabilities to understand intents.

Figure 8 displays an error analysis comparing human annotations with Llama2 predictions. Both humans and Llama2 are prone to misclassify claim and fact changes, which may stem from subjective statements being phrased in a fact-like manner and the common occurrence of intertwining both aspects within a single sentence. Llama2 demonstrates a propensity to over-predict clarity changes, often misinterpreting fact/evidence (7.84%), grammar (3.54%), and claim (3.05%) changes.

	#S	#W	Factuality	Comprehensiveness	Specificity	Compactness	Organization
human	19	346	100%	98.82%	95.56%	1.90	100% section
GPT-4	16	309	95.96%	79.09%	89.82%	2.36	72.5% action, 17.5% section

Table 4: Human evaluation and human vs. LLM comparison in document edit summarization. Demonstrated are the average counts of summary sentences (#S) and words (#W), as well as the five measures (§6.2).

6.2 Document Edit Summarization

Task formulation and data. Our full-scope annotations of document revisions enable a novel task, document edit summarization, which constitutes the foundational basis for generating author responses. This is formulated as a text generation task, given a complete list of sentence edits $e(x_i^{t+1,g}, x_j^{t,g})$ within a document, associated action and intent labels $EL_{i,j}$, as well as associated section titles. We conduct experiments on 42 documents with human-written edit summaries. These documents contain an average of 33 sentence edits, resulting in a median input length of 3,916 tokens.

Models and methods. As almost half of the inputs exceed Llama2's constraints and preliminary trials yield unsatisfactory results, we opt for GPT-4 to handle this challenging task in a zero-shot manner. We performed a human evaluation of the generated summaries, systematically comparing them with human-authored summaries across five dimensions: *factuality*, *comprehensiveness*, *specificity*, *compactness*, and *organization*. Each summary sentence a_n is linked to its respective sentence edits, creating ea linkages. And a_n is annotated if it does not refer to actual edits and labels. or it is hard to connect with specific edits. For instance, vague summaries such as 'there are grammar corrections' pose challenges in establishing precise associations. *Factuality* is quantified by the percentage of summary sentences that accurately refer to actual edits. *Comprehensiveness* denotes the extent of edits encapsulated within the summary. *Specificity* reflects the proportion of concrete summary sentences. *Compactness* is gauged by the average number of edits incorporated into a single summary sentence. And *organization* refers to the logical arrangement of the summary content.

Results and discussion. Table 4 provides a comparative analysis between human-authored and LLM-generated summaries. Humans typically produce marginally lengthier text with more sentences, ensuring impeccable factuality alongside elevated comprehensiveness and specificity. Conversely, GPT-4 fails to address 21% of document edits, exhibiting factuality concerns in 4% of summary sentences, and lacking specificity in 10% of cases. Additionally, GPT-4 summaries demonstrate a slightly higher level of compactness, averaging 2.36 edits condensed into a single sentence. While humans typically organize summaries by sections, reflecting conventional sequential reading patterns. GPT-4 also exhibits a structured logical arrangement but often organizes summaries by action labels, usually beginning with additions and deletions.

7 Conclusion

We have introduced the Re3 framework and the Re3-Sci dataset, for empirical analysis and development of NLP assistance for text-based collaboration. Through annotation study and data analysis, we have demonstrated the utility of the framework and revealed novel insights into human behavior in collaborative document revision and peer review, including relationships between specific edit actions and intents, focused localization patterns, clustering tendencies within paragraphs, and the acceptance rates for review requests. Our automation experiments have assessed the ICL and CoT capabilities of state-of-the-art LLMs on four tasks for collaborative revision assistance. In the classification tasks with Llama2-70B, we noted that using default static ICL demonstrations with CoT rationale produces satisfactory results, demonstrating the efficacy of this straightforward yet effective prompting strategy. In the document edit summarization task, GPT-4 demonstrated the ability to generate coherent summaries but faced challenges related to factuality and comprehensiveness.

Our work paves the path towards systematic full-scope study of text-based collaboration in NLP and beyond. The framework, taxonomy, annotation methods and tools are applicable to diverse domains. The dataset offers a robust foundation for multifaceted research for collaborative revision assistance. Future work may encompass tasks like identifying text segments necessitating revision and generating revisions guided by review requests or specified actions and intents.

Limitations

This study has several limitations that should be considered when interpreting our results and the implications we draw from them. From the data and modeling perspective, the study's exclusive focus on English-language scientific publications is due to the restricted availability of openly licensed source data. Studying the transferability of our findings to new languages, domains, application settings and editorial workflows is an exciting avenue for future research, which can be supported by our openly available annotation environment and protocols. Our study used human-generated edit summaries instead of author responses or summaries of changes written by the authors themselves due to the lack of data. As peer-reviewing data collection becomes increasingly popular in the NLP community, we expect new datasets to enable such studies in the future.

From a task perspective, it is important to highlight that the implementations and results presented in this study serve as illustrations of the proposed tasks. Their primary purpose is to ascertain the technical feasibility and lay the groundwork for the development of future NLP systems for collaborative writing and revision assistance. Consequently, the provided implementations have inherent limitations. For instance, our approach selectively utilizes state-of-the-art LLMs without conducting comprehensive comparisons with other LLMs or smaller fine-tuning-based models. A systematic exploration of NLP approaches for the proposed tasks lies beyond our scope and is left for the future.

Ethics Statement

The analysis of text-based collaboration and the corresponding NLP assistance applications have a potential to make knowledge work more efficient across many areas of human activity. We believe that the applications and analysis proposed in this paper deliver equitable benefits to every stakeholder involved in the procedure – authors, co-authors, reviewers, and researchers who want to study their collaborative text work.

The human annotators employed in our study were fairly compensated with a standard salary for student assistants in the country of residence. They were informed and consented to the publication of their annotations as part of this study. The annotation process does not entail the gathering or handling of their personal or sensitive information. For privacy protection, both author metadata and annotator identities have been omitted from the data release.

Both subsets of the source data are licensed under CC-BY-NC 4.0, ensuring that the construction and use of our dataset comply with licensing terms. Our annotated Re3-Sci dataset is available under a CC-BY-NC 4.0 license.

References

Jianzhu Bao, Bin Liang, Jingyi Sun, Yice Zhang, Min Yang, and Ruifeng Xu. 2021. Argument pair extraction with mutual guidance and inter-sentence relation graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3923–3934, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Liying Cheng, Lidong Bing, Qian Yu, Wei Lu, and Luo Si. 2020. APE: Argument pair extraction from peer review and rebuttal via multi-task learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7000–7011, Online. Association for Computational Linguistics.

Mike D'Arcy, Alexis Ross, Erin Bransom, Bailey Kuehl, Jonathan Bragg, Tom Hope, and Doug Downey. 2023. Aries: A corpus of scientific paper edits made in response to peer reviews.

Johannes Daxenberger and Iryna Gurevych. 2012. A corpus-based study of edit categories in featured and non-featured Wikipedia articles. In *Proceedings of COLING 2012*, pages 711–726, Mumbai, India. The COLING 2012 Organizing Committee.

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

Nils Dycke, Ilia Kuznetsov, and Iryna Gurevych. 2023. NLPeer: A unified resource for the computational study of peer review. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5049–5073, Toronto, Canada. 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.

Kraig Finstad. 2010. The usability metric for user experience. *Interacting with Computers*, 22:323–327.

Michael Fromm, Evgeniy Faerman, Max Berrendorf, Siddharth Bhargava, Ruoxia Qi, Yao Zhang, Lukas Dennert, Sophia Selle, Yang Mao, and Thomas Seidl. 2020. Argument mining driven analysis of peer-reviews. In *AAAI Conference on Artificial Intelligence*.

Yang Gao, Steffen Eger, Ilia Kuznetsov, Iryna Gurevych, and Yusuke Miyao. 2019. Does my rebuttal matter? insights from a major NLP conference. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1274–1290, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinyu Hua, Mitko Nikolov, Nikhil Badugu, and Lu Wang. 2019. Argument mining for understanding peer reviews. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2131–2137, Minneapolis, Minnesota. Association for Computational Linguistics.

Chao Jiang, Wei Xu, and Samuel Stevens. 2022. arXivEdits: Understanding the human revision process in scientific writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9420–9435, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Omid Kashefi, Tazin Afrin, Meghan Dale, Christopher Olshefski, Amanda Godley, Diane Litman, and Rebecca Hwa. 2022. ArgRewrite v.2: an annotated argumentative revisions corpus. *Language Resources and Evaluation*, 56(3):881–915.

Neha Kennard, Tim O'Gorman, Rajarshi Das, Akshay Sharma, Chhandak Bagchi, Matthew Clinton, Pranay Kumar Yelugam, Hamed Zamani, and Andrew McCallum. 2022. DISAPERE: A dataset for discourse structure in peer review discussions. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1234–1249, Seattle, United States. Association for Computational Linguistics.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico.

Ilia Kuznetsov, Jan Buchmann, Max Eichler, and Iryna Gurevych. 2022. Revise and Resubmit: An Intertextual Model of Text-based Collaboration in Peer Review. *Computational Linguistics*, 48(4):949–986.

J Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33 1:159–74.

Vladimir I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

M. Parker. 2017. *Digital Signal Processing 101: Everything You Need to Know to Get Started*. Elsevier Science.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Qian Ruan, Malte Ostendorff, and Georg Rehm. 2022. HiStruct+: Improving extractive text summarization with hierarchical structure information. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1292–1308, Dublin, Ireland. Association for Computational Linguistics.

Alexander Spangher, Xiang Ren, Jonathan May, and Nanyun Peng. 2022. NewsEdits: A news article revision dataset and a novel document-level reasoning challenge. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 127–157, Seattle, United States. Association for Computational Linguistics.

Chenhao Tan and Lillian Lee. 2014. A corpus of sentence-level revisions in academic writing: A step towards understanding statement strength in communication. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 403–408, Baltimore, Maryland. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, D. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, A. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. Korenev, Punit Singh Koura,

Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Huichao Xue and Rebecca Hwa. 2014. Redundancy detection in ESL writings. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 683–691, Gothenburg, Sweden. Association for Computational Linguistics.

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, Rebecca Hwa, Diane Litman, and Homa B. Hashemi. 2016. ArgRewrite: A web-based revision assistant for argumentative writings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 37–41, San Diego, California. Association for Computational Linguistics.

A Label Taxonomy and Examples

Table 12 presents examples of revisions analyzed according to the three dimensions: granularity, action, and intent, illustrating their importance and indispensability. Table 13 and Table 14 offer detailed definitions and examples for the edit action labels and the edit intent labels, respectively.

B Annotation

B.1 Sentence Segmentation

Both F1000RD and NLPeer datasets contain structured documents as intertextual graphs (ITG), a comprehensive document representation format that maintains document structure, cross-document links, and granularity details (Kuznetsov et al., 2022). In those ITGs, paragraphs are the most refined text elements. For our study, we opt to commence with more granular units of sentences. This creates a solid baseline for subsequent expansion to broader units, or to microscopic subdivisions.

We augment the original ITG documents with sentence nodes, employing an assembled sentence segmentation methodology using spaCy[3] and ScispaCy[4] (Neumann et al., 2019). In our preliminary testing, we discovered that neither spaCy nor ScispaCy sentence splitters are infallible for segmentation, with neither consistently outperforming the other. They can erroneously segment text based on punctuation, such as dots, which are critical for accurate revision alignment. For instance, a dot within numerical values in a sentence could trigger an incorrect segmentation and result in two sentence units. If this dot is omitted in the new version, the sentence is correctly extracted, leading to significant challenges and errors in aligning the two sentences as a revision pair. We employ an assembly of the two sentence splitters, opting for fewer segmentations yielding a smaller number of longer sentences, which mitigates most incorrect splits. Additionally, special nodes such as article titles, section titles, and list elements are not split.

The segmentations are verified and corrected by a linguistics expert, demonstrating that this integrated approach significantly enhances accuracy compared to using either splitter individually.

B.2 Sentence Revision Pre-alignment

Identifying revision pairs from two lengthy documents is challenging, especially complicated by the expansive scope for comparison and the presence of recurring content. In lengthy documents, it's crucial to align similar sentences, but it's even more vital to avoid aligning non-relevant sentences with overlapping content. To address this, we design a lightweight algorithm to automatically pre-annotate sentence revision pairs, additions and deletions, which achieves a decent accuracy of 0.95. The algorithm is detailed in Algorithm 1 which follows these steps:

1. For sentence-level alignment ($g = S$), after removing identical paragraph pairs, followed by the removal of identical sentence pairs from the remaining text, there remain k sentences in the new document D^{t+1} and l sentences in the old document D^t.
2. For each potential pair, a set of similarity measures $m \in M$ is computed, including

[3] Version 3.2.4

[4] We use the implementation provided at: https://github.com/allenai/scispacy/blob/main/scispacy/custom_sentence_segmenter.py

Levenshtein distance (Levenshtein, 1965) and fuzzy string matching[5], as well as semantic similarity measured by SBERT (Reimers and Gurevych, 2019).

3. For each remaining sentence $x_i^{t+1,g} \in D^{t+1}$, using each measure *m*, the algorithm identifies the most similar candidate $x_j^{t,g}$. If the similarity score exceeds threshold *t1* and all other similarity scores between $x_i^{t+1,g}$ and $x_j^{t,g}$ surpass *t0*, $x_j^{t,g}$ is included in the candidate list C_i, ensuring pairs similar in **both** form and meaning are found.
4. The function *f* determines the most frequent element in the resulting candidate list C_i. In cases of multiple equally frequent elements due to repeated content, the alignment is assigned to the candidate closest in location, determined by

$$d_{i,j} = \left| \frac{p_i}{\#P^{t+1}} - \frac{p_j}{\#P^t} \right| \quad (1)$$

where p_i is the linear index of the paragraph containing $x_i^{t+1,g}$, and $\#P^{t+1}$ is the total number of paragraphs in D^{t+1}. Similarly, p_j is the linear index of the paragraph containing $x_j^{t,g}$, and $\#P^t$ is the total number of paragraphs in D^t. For example, a sentence from the conclusion is more likely aligned with one from the final parts rather than the introduction.
5. If the candidate list is empty after step 3, the sentence $X_i^{t+1,g}$ stays unaligned, indicating its addition. Finally, if a sentence in D^t ends up unaligned to any in D^{t+1}, it is pre-annotated as a deletion.

The similarity thresholds *t0*, and *t1* are optimized in a pilot study on 20 document pairs, where the ideal configuration was determined to be *t0*=40, and *t1*=85, with a similarity of 100 indicating a perfect match.

B.3 Cross-document Annotation Interface

For the human annotation process, we utilized the INCEpTION platform (Klie et al., 2018). We developed a **cross-document environment** [6] that offers the complete context of two documents, facilitating full-document revision analysis and various cross-document annotation tasks. Figure 9 illustrates the annotation interface.

[5]We use fuzzywuzzy 0.18.0 at: `https://github.com/seatgeek/fuzzywuzzy`

[6]URL omitted for anonymity

Algorithm 1 Sentence pre-alignment algorithm

Input : $x_i^{t+1,g} \in D^{t+1}$, $x_j^{t,g} \in D^t$, g = S
Output : $alignS \in 1^{k \times l}$ to $0^{k \times l}$
Ensure : $0 < t0, t1 < 100$, $t0 < t1$

```
for i ← 1 to k do
    for j ← 1 to l do
        for m ∈ M do
            simS[m, i, j] ← m(x_i^{t+1,g}, x_j^{t,g})
        end for
    end for
end for
for i ← 1 to k do
    for m ∈ M do
        j_max = argmax_{i,m} simS[m, i, j]
        if simS[m, i, j_max] > t1
and all(simS[m, i, j_max] > t0, m ∈ M) then
            C_i ← C_i + j_max
        end if
    end for
    if len(f(C_i)) == 1 then
        j_align = f(C_i)[0]
        alignS[i, j_align] = 1
    else if len(f(C_i)) > 1 then
        j_align = argmin_i d(i, j), j ∈ f(C_i)
        alignS[i, j_align] = 1
    end if
end for
```

We posit that presenting only two isolated sentences without full document context is insufficient for thorough long document revision annotation. In long papers, crucial content often recurs in sections like the abstract, introduction, and conclusion, making document structure and context essential for accurate revision alignment. Context also plays a significant role in analyzing revision intent. For instance, if the authors change the name of their proposed method, annotators might perceive it as a different method and label it as a semantic change when only given two sentences. However, with the full document context, annotators can recognize a consistent name change throughout the paper, understanding that the referred method remains the same, thus categorizing it as a change for clarity.

B.4 Iterative IAA Assessment

Table 5 demonstrates that the IAA has progressively improved after implementing iterative quality management between data packages, thereby evidencing the efficacy of the employed strategy.

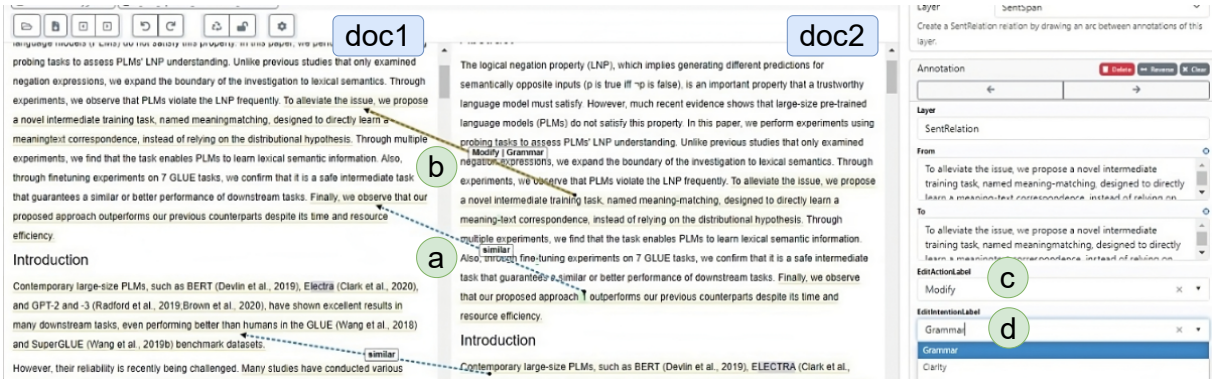


Figure 9: INCEpTION enables cross-document annotation in the context of two full documents (doc1-doc2). Annotators can scroll up and down to read the entire document context. The document's structure, including its sections and paragraphs, is preserved. The results from the pre-alignment algorithm (§B.2) are provided (a). Annotators are tasked to validate the pre-alignments (b), and select the edit action (c) and intent (d) labels.

Tasks ↓ / +Data packages →	Val.set	+DP1	+DP2	+DP3
S label	0.40	0.75	0.77	**0.78**
S align	0.99	1	1	**1**

Table 5: Inter-annotator agreement measured by Krippendorf's α on accumulative data packages, which are improved through iterative quality management. S align: sentence edit alignment, S label: sentence edit labeling.

B.5 Annotators and Tasks

For the development of the **annotation environment, taxonomy, and guidelines**, we recruited three annotators: one with expertise in computer science and two specializing in linguistics. The components were iteratively refined based on the annotators' feedback, and further validated in a pilot study with 20 documents, ensuring the robustness and applicability of our methodologies in a practical setting.

For the **subsequent annotation**, six master's students possessing C1-level English proficiency were recruited, including two in-house annotators who contributed to the prior development. Among these annotators, four specialize in linguistics, one has a background in computer science and one in engineering. For the sentence-level edit alignment and labeling tasks, each sample is annotated by three annotators, including one in-house annotator among them. A perfect IAA of 1 Krippendorff's α for alignment and a substantial IAA of 0.78 α for labeling was achieved (§B.4). For tasks at the subsentence level, a linguistics expert verified and annotated the edit spans and alignments, which were subsequently labeled for action and intent by three annotators, resulting in an IAA of 0.76 α. In the tasks of extracting and labeling review requests (refer to RQ5 in §5), eight documents were annotated by two annotators, achieving an IAA of 0.75 α. Given the substantial IAA, the remaining documents were evenly distributed and annotated individually by each annotator. Likewise, the task of associating review requests with their corresponding revisions was undertaken by two annotators on eight documents, resulting in an IAA of 0.68 α. Subsequent documents were distributed and each was handled by a single annotator. Each of the six annotators participated in generating edit summaries, with one annotator assigned per document. They then linked the summary sentences to the corresponding edits.

The **human evaluation** of the LLM-generated document edit summaries was conducted by the first author of this study, an NLP researcher with expertise in both linguistics and computer science. This researcher also assessed the factuality and specificity of the human-authored summaries (§6.2).

B.6 Annotator Survey

Following annotation, annotators complete a survey evaluating the effectiveness and efficiency of the guideline, training process, label taxonomy, cross-document annotation environment, and the annotation tool's usability. They are presented with various statements and asked to rate their agreement or disagreement on a 7-point scale, where 1 signifies 'Strongly Disagree' and 7 'Strongly Agree'.

Table 6 indicates that annotators find the guidelines, label taxonomy, and annotation interface adequate, with the iterative training process being

Survey Questions	Avg. Score
Guideline	
The annotation tasks in the guideline are clear to me.	6.8
The label schema, the definitions and the examples in the guideline are clear to me.	6.4
The *annotation procedure* in the guideline is clear to me.	6.8
With the guideline and the annotation demos, I know how to use the annotation tool to accomplish the annotation tasks.	7.0
Training	
The test annotation, discussion and correction on the first validation set have improved my understanding of the tasks and the labels.	6.2
Discussions on my individual questions have improved my understanding of the tasks and the labels.	6.8
Discussions on the summarized common mistakes have improved my understanding of the tasks and the labels.	6.8
Over time, my uncertain samples and questions have decreased significantly.	6.8
Label taxonomy	
The edit action labels can completely cover the edit actions seen.	6.8
The edit intent labels can completely cover the edit intents seen.	6.2
Cross-document annotation environment	
For *edit alignment and action labeling*, the cross-document context is crucial.	6.8
For *edit intent labeling*, the cross-document context is crucial.	6.4
Challenge	
It is hard to detect alignment and label the edit action.	2.2
It is hard to detect the actual differences of a revision pair.	3.8
It is hard to label the edit intent.	4.0
Usability of the annotation tool	
The tool's capabilities meet my requirements.	5.8
Using the tool is a frustrating experience.	2.8
The tool is easy to use.	5.2
I have to spend too much time correcting things with the tool.	1.8
Average UMUX score	77 ± 15

Table 6: Annotator survey. Annotators are presented with the statements and are asked to rate their level of agreement or disagreement on a 7-point scale, where 1 represents 'Strongly Disagree' and 7 represents 'Strongly Agree'. The final section displays a UMUX survey to measure the usability of the annotation tool and the average system UMUX score.

particularly effective. They highlight the value of discussions on individual queries and common mistakes as the most beneficial aspect of the training. Regarding the taxonomy, annotators report that the existing taxonomy adequately encompasses all observed revisions, and they did not feel the need for additional labels during the annotation process. Additionally, they recognize the importance of the cross-document annotation environment, especially in aligning revision pairs and labeling edit actions. This is also evident in their assessments of the subtasks' difficulty, with the greatest challenges being change detection and intent identification. Moreover, they perceive the annotation tool as highly usable, as indicated by the UMUX (Finstad, 2010) survey and the average system UMUX score.[7]

Add	Delete	Modify	Merge	Split	Fusion
28.93	12.44	**54.54**	1.53	2.3	0.26

(a) Proportions (%) of each edit action label.

Grammar	Clarity	Fact/Evidence	Claim	Other
14.38	21.75	**45.03**	15.43	2.7

(b) Proportions (%) of each edit intent label.

Table 7: Edit action and intent distributions.

[7]The average system UMUX score is calculated according to: https://blucado.com/understanding-the-umux-a-guide-to-the-short-but-accurate-questionnaire/

C Dataset analysis

Table 7 demonstrates the proportions of each edit action or intent label. Figure 10 illustrates the distribution of both the document edit ratio and the semantic document edit ratio. Figure 11 displays the distribution of the Crest Factor (CF) for the documents, measured at the paragraph and section levels.

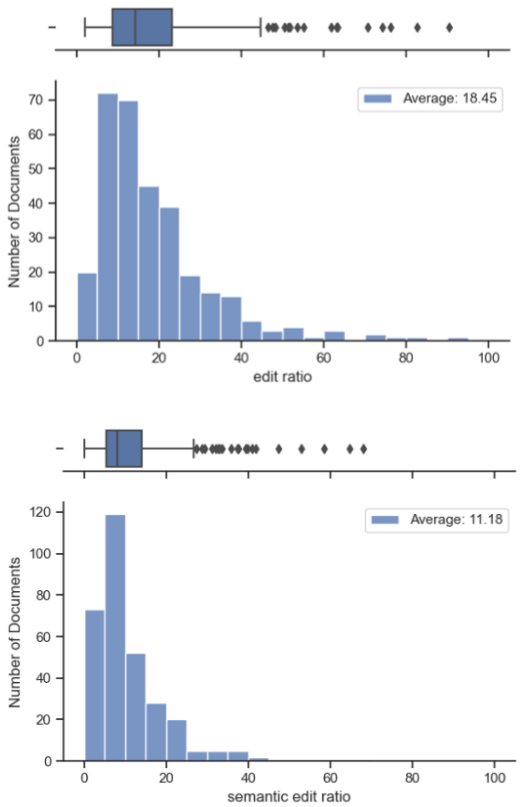


Figure 10: Document edit ratio and semantic edit ratio (%). The y-axis denotes the number of documents.

D Automation with LLMs

D.1 Edit intent classification

Table 11 shows an example prompt with system instruction, demonstration and task instruction used for experiments. Table 8 presents the performance of Llama2-70B in identifying edit intents for additions and deletions. Echoing the findings in Table 3 on revision pairs, we observe that default examples with CoT reasoning yield strong results. These outcomes are further enhanced when three demonstrations selected via the *diff* method are included. Table 10 presents the results of the joint evaluation conducted on all 8,937 test samples. The challenge

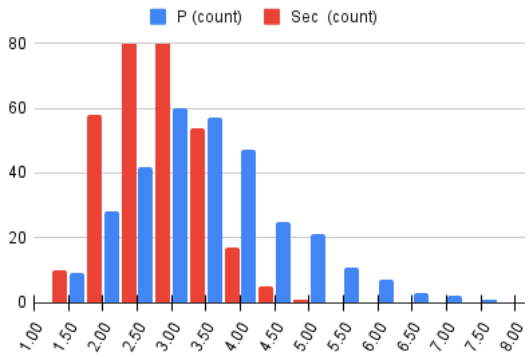


Figure 11: The Crest Factor (CF) of the documents calculated at both paragraph and section levels (P/Sec). The x-axis represents the CF value, while the y-axis shows the count of documents. The CF is a measure of the peak amplitude in a distribution, with an even distribution corresponding to CF=1. For each document, the CF is determined using a vector that denotes the count of sentence edits in each paragraph or section. For example, [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 12, 0, 0, 0, 0], CF=3.95

in identifying edit intents is particularly evident in revision pairs, highlighted by the low precision in Clarity, low recall in Fact/Evidence, and the difficulties associated with low-sourced Claim and Other classes.

D.2 Revision alignment

Task formulation and data. The task is conceptualized as a binary classification problem, where the goal is to determine if a given pair of sentences $x_i^{t+1,g} \in D^{t+1,g}$, $x_j^{t,g} \in D^t$ constitutes a revision pair $e(x_i^{t+1,g}, x_j^{t,g})$. Along with the 6,353 revision pairs from the Re3-Sci dataset, an equivalent number of negative samples are created, resulting in a total of 12,706 samples for experimental purposes. 80% are used for testing and 20% for training. To preserve the task's complexity, negative samples are composed by pairing revised sentences within the same document that do not link to each other but likely address similar topics. This simulates the intricate nature of revisions in lengthy documents as detailed in §B.2.

Models and methods. For this task, we employ the Llama2-70B model and apply the same ICL and CoT methods used for the edit intent classification task, as detailed in §6.1.

Results and discussion. Mirroring the same findings observed in the edit intent classification tasks, Table 9 also indicates that Llama2-70B does not solely rely on the provided demonstrations. This is evident from the observation that, although the ma-

Baselines			
Random	0.34	Majority	0.66
diff3-maj	0.73	cat3-maj	0.72
diff5-maj	0.75	cat5-maj	0.74
diff8-maj	0.76	cat8-maj	0.76
Our Models			
ICL & CoT			
①+ *dynamic examples*			
+diff3	0.74	+cat3	0.73
②+ *static examples*			
+def3-(L,R)	0.79		
③+ *def3-(L,R) + dynamic*			
+diff3	**0.83**		
④+ *dynamic + def3-(L,R)*			
+diff3	**0.83**	+cat3	0.80
+diff5	0.82	+cat5	0.80

Table 8: Llama2-70B accuracy in edit intent classification on addition and deletion samples. Baselines are assessed on the full test set, subsequent models are evaluated on 20% of the test set for validation. Scores underlined represent the best accuracy within the same method block, with the highest accuracy in bold. The numbers in the model names indicate the number of selected demonstrations. As detailed in Section 5, since additions and deletions are exclusively associated with Fact/Evidence, Claim, and Other, we use three default examples.

Baselines			
Random	0.50		
diff3-maj	0.59	cat3-maj	0.74
diff5-maj	0.58	cat5-maj	0.73
Our Models			
ICL & CoT			
①+ *dynamic examples*			
+diff3	0.95	+cat3	0.94
②+ *static examples*			
+def2-(L,R)	**0.97**	+def2-(R,L)	0.95
③+ *def2-(L,R) + dynamic*			
+diff3	0.96		
④+ *dynamic + def2-(L,R)*			
+diff3	**0.97**	+cat3	**0.97**
+diff5	0.96	+cat5	**0.97**

Table 9: Llama2-70B accuracy in revision alignment. Baselines are assessed on the full test set, subsequent models are evaluated on 20% of the test set for validation. Scores underlined represent the best within the same method block, with the highest accuracy in bold. The numbers in the model names indicate the number of selected demonstrations.

jority baselines using the examples selected by the *cat* method significantly outperform the *diff* baselines, the difference in ICL is small. Furthermore, the results also demonstrate that providing the same default examples with CoT reasoning throughout the experiments can yield favorable performance, highlighting its efficacy as a straightforward yet effective prompting strategy. Using this strategy, we achieve an accuracy of 0.97 on the full test set.

It is worth noting that our proposed pre-alignment algorithm (§B.2) achieves a strong accuracy of 0.95, with a recall of 0.99 for non-alignment and a precision of 0.99 for alignment. However, the precision for non-alignment (0.89) and the recall for alignment (0.92) are relatively lower. This discrepancy can be attributed to the utilization of high similarity thresholds and stringent aligning rules in the algorithm. In contrast, when automated with Llama2, we achieve a precision of 0.99 for non-alignment and a recall of 0.99 for alignment, which constitutes a perfect enhancement to the pre-alignment algorithm. For revision alignment, we thus propose **a two-stage approach** that combines the lightweight pre-alignment algorithm with Llama2 In-Context learning. The lightweight algorithm efficiently identifies candidates and accurately extracts revision pairs with minimal computational cost. Subsequently, we apply the proposed prompting strategy with Llama2 selectively to the non-aligned candidates, thereby identifying missing revision pairs without significantly increasing computational overhead.

D.3 Review request extraction

Task formulation and data. The task is framed as a binary classification problem, aiming to ascertain whether a particular review sentence $c_k \in C$ could instigate revisions and necessitate further processing in the revision workflow. The experimental data comprises 1,000 samples, including 560 review requests (including explicit and implicit edit suggestions, and general weakness comments) from the Re3-Sci dataset, plus 440 negative samples extracted from the same review documents. Of these, 80% are for testing and 20% for training.

Models and methods. For this task, we utilize Llama2-70B with the same ICL and CoT methods previously applied, as elaborated in §6.1.

Results and discussion. Employing the straightforward *def* method with CoT reasoning, which involves two static default demonstrations, yields an accuracy of 0.80 on the full test set. This approach achieves a high precision of 0.95 for negative samples and a remarkable recall of 0.98 for positive samples. Nevertheless, the precision for positive

class/	Total		Grammar			Clarity			Fact/Evidence			Claim			Other		
count	8937		1309			1838			4110			1432			248		
metrics	Acc.	M. F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Baselines																	
Random	0.20	0.18	0.15	0.20	0.17	0.21	0.20	0.20	0.46	0.20	0.28	0.17	0.21	0.19	0.03	0.21	0.05
Majority	0.46	0.13	0	0	0	0	0	0	0.46	1	0.63	0	0	0	0	0	0
Our Model (ICL&CoT)																	
+ diff3 + def-(L,R)																	
joint	0.7	0.69	0.79	0.72	0.75	0.54	0.85	0.66	0.85	0.68	0.76	0.61	0.58	0.6	0.76	0.62	0.69
A,D/ count	3891		0			0			2580			1135			176		
	0.78	0.77	-	-	-	-	-	-	0.86	0.8	0.83	0.62	0.73	0.67	0.85	0.76	0.80
R/ count	5046		1309			1838			1530			297			72		
	0.65	0.48	0.79	0.72	0.75	0.54	0.85	0.66	0.82	0.48	0.61	0.14	0.01	0.01	0.46	0.29	0.36

Table 10: Edit intent classification, joint evaluation of the optimal configuration using Llama2-70B on all test samples. R: revision pairs, A,D: additions and deletions. Displayed are the accuracy (Acc.), macro average F1 score (M. F1), and precision (P), recall (R), and F1 score for each label. The challenge in identifying edit intents is particularly evident in revision pairs, highlighted by the low precision in Clarity, low recall in Fact/Evidence, and the difficulties associated with low-sourced Claim and Other classes.

samples is relatively lower at 0.74, highlighting the method's inherent challenges. Future research could expand this task into a four-class classification, differentiating various types of review requests. This approach could further elucidate the methods' capabilities and limitations.

D.4 Computational details

In our classification tasks with Llama2-70B, covering revision alignment, edit intent classification, and review request extraction, we employed two RTX™ A6000 GPUs, each equipped with 48GB of memory. The batch size for inference was established at four. For the document edit summarization task using GPT-4, we processed 282,964 input tokens and produced 36,341 output tokens in total, resulting in a total expense of 3.92 US dollars.

System instruction:
You are a helpful, respectful and honest revision analysis assistant. You will read two versions of texts. Your task is to analyze the revision intent behind the difference between the two texts. The intent can be one of the following labels: fix grammar (Grammar), improve clarity (Clarity), change claim or statement (Claim), change factual information (Fact/Evidence). Grammar and Clarity are more about surface language improvements, while Fact/Evidence and Claim are more about meaning changes. If none of the above labels are relevant, please answer with 'Other'.
Demonstration with gold label and CoT rationale:
The old text is: Empirical studies on the datasets across 7 different languages confirm the effectiveness of the proposed model. The new text is: Empirical studies on the three datasets across 7 different languages confirm the effectiveness of the proposed model. LABEL: Fact/Evidence REASON: 'Three' is added to the new text. This is an addition of factual information that the empirical studies are conducted on 'three' datasets, thus the label is Fact/Evidence.
Task instruction:
Read the following old and new texts. What is the intent of the revision? Please answer with one of the labels: Grammar, Clarity, Claim, Fact/Evidence and Other. Please always answer with the template and fill the template with your answer without additional texts: LABEL:<your answer> REASON:<your answer>.

Table 11: Example of a Llama2 prompt for edit intent classification: The prompt encompasses the system instruction, demonstration examples with gold labels and optional CoT rationale, as well as the task instruction.

Example	Revision description	Notation
These findings constitute the first evidence that using our taxonomy could result in robust methods , even though more data and research seem necessary to get there .	A **subsentential** text element, i.e., the highlighted clause, is *added* for a more cautious view, *claiming* that further data and research are required. If viewed at the **sentence** level, this reflects a *modification* of an existing sentence.	(**SS**, *Add, Claim*) or (**S**, *Modify, Claim*)
... is a medication for smoke cessation. All these cases pose challenges to state-of-the-art language models. Recent work ...	An entire new **sentence** is *added* to make a *claim*. If viewed at the **paragraph** level, this represents a *modification* of an existing paragraph.	(**S**, *Add, Claim*) or (**P**, *Modify, Claim*)
The values were compared using the Bonferroni test post hoc. Also, the population density of each zone was calculated ...	*Addition* of one entire new **paragraph** to furnish *factual details*. From a **sentence-level** view, this equates to multiple sentence *additions*.	(**P**, *Add, Fact/Evidence*) or multiple (**S**, *Add, Fact/Evidence*)
However, the problem is that the hypothesis has limitations in reflecting a word's meanings~~.~~ , because w ~~W~~ords having different or even opposite meanings can appear in similar contexts.	An existing **paragraph** is *modified* to update claims. Upon closer inspection at the **sentence** level, it involves *merging* two sentences for *clarity* and *modifying* one of them to update *claims*.	(**P**, *Modify, [Clarity, Claim]*) or (**S**, *Merge+Identical, Clarity*) (**S**, *Merge+Modify, Claim*)

Table 12: Revision examples described and notated by the three revision dimensions: granularity, action and intent. The same revisions are perceived differently based on varying levels of granularity, making the three dimensions necessary for a precise analysis. Texts with strikethroughs are removed, and texts highlighted in blue are added. SS: subsentence-level, S: sentence-level, P: paragraph-level.

Edit actions		Definitions	Alignment type
Content	Add	Insert an entire new text element	1-to-0
	Delete	Remove an existing text element completely	0-to-1
	Modify	Revise an existing text element by altering a portion of it, with some parts of the original text remaining unchanged.	1-to-1
Partition	Merge	Consolidate multiple text elements into a single text element	1-to-n
	Split	Distribute a single text element into multiple separate text elements	n-to-1
	Fusion	Combination of merge(s) and split(s)	m-to-n

Table 13: Edit action definitions and alignment types. The alignment refers to the new-version-to-old-version relation, the new version is the source of the alignment and the old version is the target. Partition changes comprise a series of one-to-one alignments, each necessitating an accompanying content label (Modify or Identical) to denote whether the linked element's content has altered (also see the last example in Table 12).

Edit intents		Definitions	Subsentence-level Examples
Surface	Grammar	Correct grammatical errors, capitalization, punctuation, tense, modality, spelling, typography, abbreviations or any errors related to grammar and/or conventions to improve the language.	**Modify, Grammar:** It is freely available for ~~akademic~~ academic use.
	Clarity	Alter word choice, phrase usage, expressions and/or text format to be more formal, concise and understandable without meaning changes, or to amplify meaning for clarity.	**Modify, Clarity:** This study ~~checked out~~ examined how images affect learning.
Semantic	Fact/Evidence	Add, elaborate, extend, verify or update the fact and/or evidence from third parties, or the author's factual manipulations and observations, or delete/modify erroneous/irrelevant ones.	**Modify, Fact/Evidence:** XX, et. al. sets the state-of-the-art ROUGE result to ~~0.56~~ 0.54 .
	Claim	Change/Add/Delete the claim, statement, opinion, idea of the authors, or their overall aim of the document.	**Add, Claim:** These findings constitute the first evidence that using our taxonomy could result in robust methods , even though more data and research seem necessary to get there .
Other		Revise the text in a way that is irrelevant to any other type of edit intent.	**Add, Other** (*changes in section titles*): Experiments and Results

Table 14: Edit intent definitions and subsentence-level examples. Texts with strikethroughs are removed, and texts highlighted in blue are added.