# Are Large Language Models Good Classifiers?
# A Study on Edit Intent Classification in Scientific Document Revisions

**Anonymous EMNLP submission**


## Abstract

Classification is a core NLP task architecture with many potential applications. While large language models (LLMs) have brought substantial advancements in text generation, their potential for enhancing classification tasks remains underexplored. To address this gap, we propose a framework for thoroughly investigating fine-tuning LLMs for classification, including both generation- and encoding-based approaches. We instantiate this framework in edit intent classification (EIC), a challenging and underexplored classification task. Our extensive experiments and systematic comparisons with various training approaches and a representative selection of LLMs yield new insights into their application for EIC. To demonstrate the proposed methods and address the data shortage for empirical edit analysis, we use our best-performing model to create *Re3-Sci2.0*, a new large-scale dataset of 1,780 scientific document revisions with over 94k labeled edits. The new dataset enables an in-depth empirical study of human editing behavior in academic writing. We make our experimental framework, models and data publicly available.[1]


## 1 Introduction

Generative large language models (LLMs) have demonstrated substantial advancements in text generation tasks (Zhang et al., 2023; Wang et al., 2023; Pham et al., 2023). However, their potential for enhancing classification tasks, a significant subset of NLP applications, remains underexplored. The predominant strategy for applying LLMs to classification tasks is to cast them as generation tasks, followed by instruction tuning (Qin et al., 2023; Sun et al., 2023; Peskine et al., 2023; Milios et al., 2023; Patwa et al., 2024), supervised fine-tuning (Parikh et al., 2023), and active learning (Rouzegar and Makrehchi, 2024), all of which aim to generate label strings within the output tokens. Recent studies (Lee et al., 2024; Kim et al., 2024; Meng et al., 2024) have shown the superiority of LLMs as embedding models on the MTEB benchmark (Muennighoff et al., 2023). However, there is a lack of a holistic framework for a systematic study of the classification capabilities of LLMs in end-to-end fine-tuning paradigms. Yet, such a framework is important as it extends beyond the current use of LLMs as generative or embedding models for classification, opens new opportunities for a wide range of real-world tasks, and reveals novel potential for advanced LLM training and utilization.

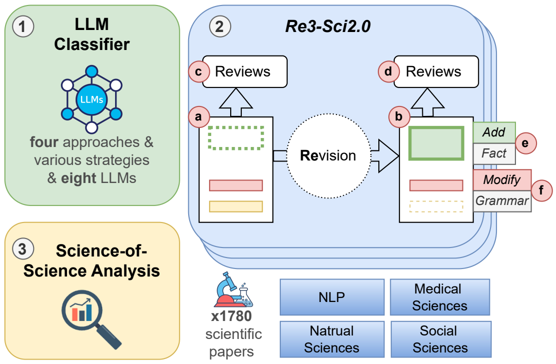



Figure 1: In this work, we (1). present a general framework to explore the classification capabilities of LLMs, conducting extensive experiments and systematic comparisons on the EIC task; (2). use the best model to create the *Re3-Sci2.0* dataset, which comprises 1,780 scientific document revisions (a-b), associated reviews (c, d), and 94,482 edits annotated with action and intent labels (e, f), spanning various scholarly domains; (3). provide a first in-depth empirical analysis of human editing behavior using this new dataset.

To instantiate the framework, we seek a **complex**, **challenging**, and **underexplored** task that is **crucial** for addressing unresolved real-world applications. Edit intent classification (EIC) is such a complex task, aiming to identify the purpose of textual changes, necessitating a deep understanding of the fine-grained differences between paired in-

[1] URL omitted for anonymity

puts. Previous works have provided small human-annotated datasets and demonstrated the crucial role of the intent labels in studying domain-specific human editing behavior (Zhang et al., 2016; Yang et al., 2017; Kashefi et al., 2022; Ruan et al., 2024). However, due to the high cost of human annotation, existing datasets are limited in size. There is a lack of effective NLP automation and extensive labeled datasets to facilitate larger-scale revision analysis. From the modeling perspective, previous studies have primarily explored EIC using basic feature engineering (Zhang et al., 2016; Yang et al., 2017; Kashefi et al., 2022), fine-tuning small pre-trained language models (PLMs) (Du et al., 2022; Jiang et al., 2022), or instruction tuning with LLMs (Ruan et al., 2024). Advanced methodologies involving fine-tuning LLMs remain unexplored. The suboptimal results of previous works (Table 1) further highlight the task's inherent difficulty and the necessity for advancements in NLP.[2]

To close the gap, we introduce a general framework to explore the use of LLMs for classification, featuring one generation-based and three encoding-based fine-tuning approaches (§3). We instantiate the framework in EIC, conduct extensive experiments and provide novel insights from systematic comparisons of the four approaches, eight LLMs, and various training strategies. Our findings reveal that partially fine-tuned LLMs exhibit superior encoding and classification capabilities on EIC compared to fully fine-tuned PLMs and instruction-tuned larger LLMs. We also identify the most effective approach and LLM, among other insights (§4). To illustrate the application of the proposed methods and address the lack of data for extensive edit analysis, we use our models to create *Re3-Sci2.0*, a large-scale dataset with 1,780 scientific document revisions and 94,482 labeled edits across various research domains (§5). This dataset enables the first in-depth science-of-science (Fortunato et al., 2018) analysis of scientific revision success and human editing behavior across research domains (§5.3). Our work thus makes four key **contributions**:

- A general framework for fine-tuning LLMs for classification tasks, with four approaches and various training strategies.
- Extensive experiments on EIC, and systematic comparisons of different approaches, training strategies, PLMs and LLMs.
- A large dataset of 1,780 scientific document revisions with 94,482 edits, annotated by our best model, which achieves a macro average F1 score of 84.3.
- A first in-depth science-of-science analysis of scientific revision success and human editing behavior across various scholarly domains.

Our work paves the path towards systematically investigating the use of LLMs for classification tasks. Our experiments yield substantial results in the challenging EIC task. The resulting large-scale dataset facilitates empirical analysis of human editing behavior in academic publishing and beyond.

[2]Note that direct performance comparison is not possible due to different datasets, label sets and data sizes, but they illustrate the inherent difficulty of EIC despite data variations.

## 2 Related Work

| | #label | #train | #test | acc. | method |
|---|---|---|---|---|---|
| Zhang et al. (2016) | 8 | 1,757 | 10CV | 58.8* | FE |
| Yang et al. (2017) | 13 | 5,777 | 10CV | 59.7* | FE |
| Kashefi et al. (2022) | 9 | 3,238 | 5CV | 68 | FE |
| Du et al. (2022) | 5 | 3,254 | 364 | 49.4* | PLM |
| Jiang et al. (2022) | 4 | 600 | 200 | 84.4 | PLMs |
| Jiang et al. (2022) | 9 | 600 | 200 | 79.3 | PLMs |
| Ruan et al. (2024) | 5 | 2,234 | 8,936 | 70 | LLM (inst) |
| Ours | 5 | 7,478 | 2,312 | 85.6 | PLMs & LLMs |

Table 1: Comparison of related works on EIC, including counts of unique intent labels, training and test samples, best accuracy (or *macro average F1 scores), and explored methods. nCV: n-fold cross-validation. FE: feature engineering.

**Edit Intent Classification.** Identifying the underlying intent of textual edits is a challenging yet underexplored task, with only a few studies contributing taxonomies, datasets and methodologies. Among these, several works (Zhang et al., 2016; Yang et al., 2017; Kashefi et al., 2022) have investigated various feature engineering techniques and employed basic classifiers such as SVM (Cortes and Vapnik, 1995), MULAN (Tsoumakas et al., 2011), and XGBoost (Chen and Guestrin, 2016). Other studies (Du et al., 2022; Jiang et al., 2022) explored fine-tuning PLMs such as RoBERTa (Liu et al., 2019), T5 (Raffel et al., 2020), and PURE (Zhong and Chen, 2021). Ruan et al. (2024) is the first application of LLMs for EIC. However, it is limited to using Llama2-70B (Touvron et al., 2023) with instruction tuning, without any fine-tuning. As outlined in Table 1, our work is the first to systematically compare different fine-tuning approaches for a broad set of PLMs and LLMs using various training strategies for EIC, achieving substantial progress in this challenging task.

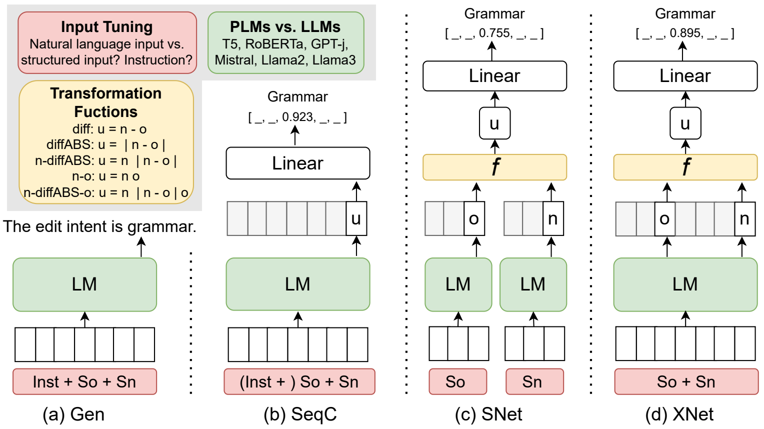



Figure 2: Proposed approaches with a systematic investigation of the key components: input types (red), language models (green), and transformation functions (yellow). See §3 and §4 for details.

**LLMs for Classification.** Previous studies have utilized LLMs for classification, primarily aiming to generate label strings within the output tokens through instruction tuning (Qin et al., 2023; Sun et al., 2023; Peskine et al., 2023; Milios et al., 2023; Patwa et al., 2024). Few studies have enhanced LLMs to generate label text through supervised fine-tuning (Parikh et al., 2023) and active learning (Rouzegar and Makrehchi, 2024). Additionally, recent studies (Lee et al., 2024; Kim et al., 2024; Meng et al., 2024) have demonstrated the superiority of LLMs as embedding models on MTEB[3] (Muennighoff et al., 2023), an extensive text embedding benchmark where embeddings are processed by additional classifiers. However, there is a lack of a holistic framework for systematically investigating the encoding capabilities of LLMs in end-to-end fine-tuning paradigms. We are the first to address the gap by proposing encoding-based methodologies that extensively investigate and fine-tune LLMs as supervised classification models, systematically comparing these methodologies with the generation-based approach within a unified framework. While this work focuses on the challenging and crucial EIC task (§1), our methodologies and the framework are applicable to a wide range of classification tasks.

## 3 Framework

We investigate four distinct approaches to fine-tune LLMs for classification (§3.1), use various training strategies including three input types (§3.2) and five transformation functions (§3.3), systematically comparing different language models (§3.4).

### 3.1 Approaches

We illustrate the proposed approaches to text classification using the EIC task. We formulate it as a multi-label classification task involving a sentence edit pair $e(S_o, S_n)$, where $S_o$ represents the original sentence and $S_n$ denotes the new sentence after the edit. In cases of sentence additions or deletions, only the single added/deleted sentence ($S_n$/$S_o$) is provided, while the corresponding pair sentence remains empty. The objective is to predict an edit intent label $l$ from a set of $k$ possible labels $L$. As illustrated in Figure 2,

- **Approach *Gen*** addresses the task as a text generation task, aiming to produce the label string within the output tokens from input text that includes the task instruction, the old sentence $S_o$, and the new sentence $S_n$.
- **Approach *SeqC*** treats the task as a sequence classification task using LLMs equipped with a linear classification layer on top. It utilizes the last hidden states of the last token ($u$) as the input embedding for classification. The linear layer transforms $u$ of the model size $d$ into a $k$-dimensional logit vector, where the maximum value indicates the predicted label.
- **Approach *SNet*** employs a Siamese architecture for sequence classification. It processes the two sentences independently through twin Siamese LLMs, producing $o$ and $n$ (representing the last token of each), for the old and new sentences respectively. A transformation function $f$ (§3.3) combines these into a single representation $u$ for classification.

[3]https://huggingface.co/blog/mteb

- **Approach *XNet*** employs a cross network to process both sentences simultaneously through a single LLM, extracting the last-token embeddings $o$ and $n$ for the old and new sentences respectively. They are then merged into a single representation $u$ by a transformation function $f$ for classification.

### 3.2 Input Tuning

The input text, indicated by red blocks in Figure 2, comprises three components: the task instruction (*inst*), the original sentence $S_o$ and the new sentence $S_n$. The task instruction outlines the task's objective and specifies the possible labels. The input text is provided in two different formats: (1) *natural input*, which includes only the content of the instruction and the sentences, and (2) *structured input*, where the content is enclosed within specific structure tokens such as *<instruction></instruction>*, *<old></old>*, and *<new></new>*. In our experiments, we tune the presence of task instructions and the input text formats to explore their effects (§4). Examples of input texts are displayed in Table 7 in §A.

### 3.3 Transformation Functions

In approaches *SNet* and *XNet*, the representations of the old and new sentences, $o$ and $n$, can be combined into a single representation $u$ using five different transformation functions $f$:

$$f_{diff} : u = n - o \tag{1}$$

$$f_{diffABS} : u = |n - o| \tag{2}$$

$$f_{n-diffABS} : u = n \oplus |n - o| \tag{3}$$

$$f_{n-o} : u = n \oplus o \tag{4}$$

$$f_{n-diffABS-o} : u = n \oplus |n - o| \oplus o \tag{5}$$

where $\oplus$ represents vector concatenation, - denotes vector subtraction, and | | indicates that absolute values are taken from the subtraction. The five transformation functions are systematically evaluated in our experiments (§4).

### 3.4 Language Models

The proposed approaches are intended for systematically investigating fine-tuning LLMs but are readily extendable to other language models (LMs). We explore eight of the most advanced LLMs: GPT-j (Wang and Komatsuzaki, 2021), Mistral-Instruct (Jiang et al., 2023), Llama2-7B and Llama2-7B-Chat (Touvron et al., 2023), Llama2-13B and Llama2-13B-Chat (Touvron et al., 2023), Llama3-8B and Llama3-8B-Instruct[4], and compare them with two small PLMs: T5 (Raffel et al., 2020) and RoBERTa (Liu et al., 2019). Details on model selection and an overview of the chosen LLMs and PLMs are provided in §A.

## 4 Results and Discussion

### 4.1 Data and Experimental Details

For our experiments, we seek a high-quality dataset with a sufficient number of samples for fine-tuning. Re3-Sci (Ruan et al., 2024) is such a dataset, which comprises 11,566 high-quality human-labeled sentence edits from 314 document revisions. We divide the dataset into training, validation, and test sets with 7,478/1,776/2,312 edits. Re3-Sci categorizes edit intents into five distinct labels: *Grammar* and *Clarity* for surface language improvements, *Fact/Evidence* and *Claim* for semantic changes in factual content or statements, and *Other* for all other cases. The task is thus formulated as a 5-label classification challenge given a sentence revision pair (§3.1). We fine-tune all linear layers of the LLMs using QLoRA (Dettmers et al., 2023). The PLMs are fully fine-tuned with all weights being directly updated. For approach *Gen*, the output token limit is set to ten. We define *Answer Inclusion Rate (AIR)* as the percentage of samples where a label string falls within the ten output tokens, regardless of correctness. Further details are provided in §B.

### 4.2 Discussion

Table 2 shows the performance of human annotators and instruction tuning baselines using GPT-4 and Llama2-70B (details in §B), as well as the performance from approaches *Gen* and *SeqC*, comparing various input types. Table 3 presents the comparative results of approaches *SNet* and *XNet*, evaluating different transformation functions. Based on these results, we address five research questions:

**RQ1: Are fine-tuned LLMs good edit intent classifiers compared to fully fine-tuned PLMs and instruction-tuned larger LLMs?** Our results suggest that LLMs can be effectively enhanced to serve as good edit intent classifiers with our optimal approaches, outperforming larger instruction-tuned LLMs and fully fine-tuned PLMs. First, we compare our best results with the baselines. Bold texts in Table 2(b) indicate that approach *SeqC* with either Llama2-13B or Llama3-8B-Instruct achieves

[4] https://github.com/meta-llama/llama3

| Baselines | | | | | | | |
|---|---|---|---|---|---|---|---|
| | size | acc. | m. f1 | AIR | acc. | m. f1 | AIR |
| Human | - | 90.2 | 89.7 | 100 | | | |
| | | zero-shot | | | ICT+CoT | | |
| GPT-4 | - | 45.5 | 37 | 99.9 | 64.8 | 60.9 | 100 |
| Llama2-70B (2024) | 70B | - | - | - | 70$^\dagger$ | 69$^\dagger$ | 100 |

**(a). *Gen***

| | | NFT Baselines | | | Fine-tuned Models | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | ① *inst + natural input* | | | ② *inst + structured input* | | |
| base LM | size | acc. | m. f1 | AIR | acc. | m. f1 | AIR | acc. | m. f1 | AIR |
| T5 | 220M | 1.2 | 1.5 | 4.8 | 79.9 | 78.1 | 100 | 78.3 (↓1.6) | 78.0 (↓0.1) | 100 |
| GPT-j | 6B | 12.6 | 9.3 | 68.9 | 32.8 | 17.5 | 97.6 | 21.2 (↓11.6) | 12.8 (↓4.7) | 86.8 (↓10.8) |
| Mistral-Instruct | 7B | 28.0$^\dagger$ | 20.0$^\dagger$ | 99.9 | 68.5 | 63.4 | 100 | 62.8 (↓5.7) | 59.2 (↓4.2) | 100 |
| Llama2-7B | 7B | 21.4 | 10.2 | 78.2 | 34.3 | 24.7 | 100 | 60.4 (↑26.1) | 39.7 (↑15.0) | 88.7 (↓11.3) |
| Llama2-7B-Chat | 7B | 12.1 | 7.2 | 85.2 | 63.0 | 49.2 | 100 | 72.4 (↑9.4) | 45.8 (↓3.4) | 88.5 (↓11.5) |
| Llama2-13B | 13B | 13.8 | 4.3 | 93.3 | 50.9 | 32.9 | 99.9 | 73.4 (↑22.5) | 56.3 (↑23.4) | 85.9 (↓14.0) |
| Llama2-13B-Chat | 13B | 0.5 | 1.6 | 2.0 | 75.5 | 72.9 | 100 | 83.6 (↑8.1) | 82.8 (↑9.9) | 100 |
| Llama3-8B | 8B | 14.0 | 11.1 | 77.8 | 79.4 | 65.9 | 95.4 | 83.3 (↑3.9) | 68.4 (↑2.5) | 99.9 (↑4.5) |
| Llama3-8B-Instruct | 8B | 12.6 | 14.4 | 47.3 | 84.1$^\dagger$ | 82.4$^\dagger$ | 100 | **84.7**$^\dagger$ (↑0.6) | **83.7**$^\dagger$ (↑1.3) | 100 |

**(b). *SeqC***

| | | NFT Baselines | | Fine-tuned Models | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | ① *natural input* | | ② *structured input* | | ③ *inst + structured input* | |
| base LM | size | acc. | m. f1 | acc. | m. f1 | acc. | m. f1 | acc. | m. f1 |
| RoBERTa | 125M | 22.5 | 7.3 | 78.4 | 75.8 | 79.8 (↑1.4) | 78.4 (↑2.6) | 78.8 (↓1) | 75.8 (↓2.6) |
| GPT-j | 6B | 16.0 | 11.2 | 81.1 | 79.2 | 81.3 (↑0.2) | 80.0 (↑0.8) | 82.2 (↑0.9) | 80.8 (↑0.8) |
| Mistral-Instruct | 7B | 15.7 | 9.1 | 83.3 | 81.9 | 52.4 (↓30.9) | 32.8 (↓49.1) | 48.8 (↓3.6) | 32.4 (↓0.4) |
| Llama2-7B | 7B | 22.4 | 14.1$^\dagger$ | 82.7 | 81.5 | 84.3 (↑1.6) | 83.3 (↑1.8) | 84.5 (↑0.2) | 83.0 (↓0.3) |
| Llama2-7B-Chat | 7B | 24.2 | 12.5 | 81.6 | 80.1 | 84.4 (↑2.8) | 82.8 (↑2.7) | 83.8 (↓0.6) | 82.1 (↓0.7) |
| Llama2-13B | 13B | 15.5 | 5.4 | 84.0 | 82.0 | 84.9 (↑0.9) | 84.1 (↑2.1) | 85.4$^\dagger$ (↑0.5) | **84.3**$^\dagger$ (↑0.2) |
| Llama2-13B-Chat | 13B | 26.9 | 13.0 | 83.0 | 81.5 | 84.2 (↑1.2) | 82.5 (↑1.0) | 85.1 (↑0.9) | 83.7 (↑1.2) |
| Llama3-8B | 8B | 35.6$^\dagger$ | 13.0 | 84.1 | 82.3$^\dagger$ | 84.2 (↑0.1) | 83.1 (↑0.8) | 46.8 (↓37.4) | 26.4 (↓56.7) |
| Llama3-8B-Instruct | 8B | 10.6 | 9.0 | 84.4$^\dagger$ | 82.2 | **85.6**$^\dagger$ (↑1.2) | **84.3**$^\dagger$ (↑2.1) | 83.4 (↓2.2) | 81.9 (↓2.4) |

Table 2: Results of human and instruction tuning baselines, approaches (a) *Gen* and (b) *SeqC*. Reported are accuracy (acc.), macro average F1 score (m. f1) and Answer Inclusion Rate (AIR) on the test set. For each base LM, we compare the performance of the non-fine-tuned model with that of models fine-tuned using different input formats, noting performance differences in parentheses. The best-performing setting for each LM is underlined, and $^\dagger$ denotes the best-performing LM within each setting. The best metrics from each approach are highlighted in bold.

the highest macro average F1 score of 84.3. This result notably exceeds the GPT-4 baselines, both in a zero-shot setting and when enhanced with ICL and CoT. It also surpasses an instruction-tuned Llama2-70B, as reported by Ruan et al. (2024). Then, we compare the results from fine-tuning LLMs and PLMs. Table 2(b) shows that using the encoding-based approach *SeqC*, all eight LLMs surpass a fully fine-tuned RoBERTa in most settings, highlighting the superior encoding capabilities of LLMs. Table 2(a) shows that using approach *Gen*, Llama2-13B-Chat, Llama3-8B, and Llama3-8B-Instruct can achieve better or comparable results to a fully fine-tuned T5. The favorable results in Table 3(d) indicate that fine-tuning via *XNet* also effectively enhances LLMs as edit intent classifiers.

**RQ2: Which LLMs are more effective as edit intent classifiers?** Overall, an analysis of the best-performing models, marked with $^\dagger$ in Tables 2 and 3, reveals that the largest 13B Llama2 models and the latest 8B Llama3 models outperform others in most cases. Using the *Gen* approach (Table 2(a)), the instruction-fine-tuned versions of LLMs consistently and substantially outperform their non-instruction-fine-tuned counterparts, which may be attributed to their improved understanding of instructions. In *SeqC* (Table 2(b)), the non-Chat versions of the Llama2 models slightly outperform their Chat version counterparts. However, Llama3-8B-Instruct outperforms Llama3-8B using *SeqC*, particularly with more complex inputs (further discussion in RQ4). In approaches *SNet* and *XNet* (Table 3), there are no substantial or consistent performance differences among the LLMs.

**RQ3: Which approach is most effective?** Overall, approach *SeqC* demonstrates superior performance, answer inclusion rate (AIR), and inference efficiency. Regarding AIR, Table 2(a) indicates that generative models encounter AIR issues even after fine-tuning. This suggests that the generation-based approach is not optimal in practice due to its lack of robustness and difficulty in control. The other encoding-based approaches achieve perfect AIR. In terms of performance, approaches *SeqC*

| (c). *SNet* | ① *diff* | | ② *diffABS* | | ③ *n-diffABS* | | ④ *n-o* | | ⑤ *n-diffABS-o* | |
|---|---|---|---|---|---|---|---|---|---|---|
| base LM | acc. | m. f1 | acc. | m. f1 | acc. | m. f1 | acc. | m. f1 | acc. | m. f1 |
| Llama2-7B | 61.5 | 60.5 | 69.7 | 69.5 | 68.5 | 68.0 | 60.8 | 58.8 | 67.7 | 68.0$^{\dagger}$ |
| Llama2-7B-Chat | 60.7 | 56.5 | 72.4 | 71.4 | 65.4 | 64.7 | 58.7 | 55.3 | 68.5$^{\dagger}$ | 67.6 |
| Llama2-13B | 62.4 | 59.3 | 73.1 | 72.4 | 67.5 | 67.2 | 61.0$^{\dagger}$ | 59.1$^{\dagger}$ | 66.0 | 67.2 |
| Llama2-13B-Chat | 63.7$^{\dagger}$ | 61.6$^{\dagger}$ | 69.4 | 69.3 | 66.9 | 66.3 | 60.4 | 57.9 | 66.0 | 65.3 |
| Llama3-8B | 61.0 | 57.4 | 70.6 | 69.8 | 69.8$^{\dagger}$ | 68.7$^{\dagger}$ | 58.6 | 56.6 | 64.8 | 63.8 |
| Llama3-8B-Instruct | 59.9 | 56.6 | **73.3**$^{\dagger}$ | **72.9**$^{\dagger}$ | 61.2 | 54.7 | 60.6 | 58.4 | 61.2 | 54.7 |

| (d). *XNet* | ① *diff* | | ② *diffABS* | | ③ *n-diffABS* | | ④ *n-o* | | ⑤ *n-diffABS-o* | |
|---|---|---|---|---|---|---|---|---|---|---|
| base LM | acc. | m. f1 | acc. | m. f1 | acc. | m. f1 | acc. | m. f1 | acc. | m. f1 |
| Llama2-7B | 83.0 | 81.4 | 84.4 | 83.1 | 84.5 | 82.8 | 83.6 | 82.2 | 83.2 | 81.6 |
| Llama2-7B-Chat | 84.3 | 83.2 | 83.6 | 81.9 | 83.6 | 82.4 | 83.3 | 81.4 | 83.2 | 81.8 |
| Llama2-13B | 84.3 | 82.7 | 84.0 | 82.7 | 85.0 | **83.9**$^{\dagger}$ | 84.4 | 83.4 | 84.6$^{\dagger}$ | 83.7$^{\dagger}$ |
| Llama2-13B-Chat | 84.3 | 82.9 | **85.2**$^{\dagger}$ | 83.7$^{\dagger}$ | 84.5 | 83.6 | 84.9 | 83.7$^{\dagger}$ | 84.6$^{\dagger}$ | 83.3 |
| Llama3-8B | 83.7 | 82.4 | 84.1 | 82.4 | 84.7 | 83.6 | 76.7 | 73.7 | 83.5 | 82.1 |
| Llama3-8B-Instruct | 84.4$^{\dagger}$ | 83.4$^{\dagger}$ | 84.5 | 83.2 | 85.1$^{\dagger}$ | 83.7 | 85.1$^{\dagger}$ | 83.7$^{\dagger}$ | 84.1 | 83.3 |

Table 3: Results of approaches (c) *SNet* and (d) *XNet*. Reported are accuracy (acc.) and macro average F1 score (m. f1) on the test set. For each base LM, we compare the performance of models fine-tuned using different transformation functions (§3.3). The best-performing setting for each LM is underlined, $^{\dagger}$ denotes the best-performing LM within each setting. The best metrics from each approach are in bold.

and *XNet* are superior. The Siamese network (*SNet*) consistently and substantially underperforms the cross network (*XNet*) when using the same LLMs and transformation functions (Table 3). Inference efficiency is measured by the number of samples processed per second during inference. This metric is particularly important when applying the model to large datasets. Figure 5 in §C compares the three metrics for the four approaches using Llama2-13B as the base LM. Approach *SeqC* achieves perfect AIR, the best performance, and a 12x inference speedup compared to approach *Gen* and a 4x speedup compared to *SNet* and *XNet*.

**RQ4: What are the effects of the input types?** Now, we examine the ablation results detailed in parentheses in Table 2. Table 2(a) shows that using structured input instead of natural language input improves performance for the Llama2 models in approach *Gen*, though it may decrease AIR. However, for GPT-j and Mistral-Instruct, structured input has a substantial negative impact. Table 2(b) shows that in approach *SeqC*, using structured inputs positively impacts RoBERTa and all LLMs except for Mistral-Instruct. Adding the task instruction to structured inputs has minimal effects on most models, however, it particularly negatively impacts Llama3-8B.

**RQ5: What are the effects of the transformation functions?** We examine the most effective transformation functions, indicated by the most frequently underlined columns in Table 3. Table 3(c) indicates that when using *SNet*, $f_{diffABS}$ substantially outperforms all other functions across all LLMs. When using *XNet*, the best-performing functions are $f_{n-diffABS}$, $f_{diffABS}$ and $f_{diff}$, as shown in Table 3(d). However, the differences across the transformation functions are not substantial.

## 5 Application: *Re3-Sci2.0*

The original Re3-Sci dataset contains only 314 documents covering limited research domains, thus constraining in-depth science-of-science analysis of how humans improve scientific quality through revisions and how their document-based editing behavior varies across domains. Having determined the optimal approach for EIC among the considered ones, we apply our best-performing model to create *Re3-Sci2.0*: the first large-scale corpus of academic document revisions for edit analysis across research domains.

### 5.1 Data Collection and Labeling

Re3-Sci is built upon F1000RD (Kuznetsov et al., 2022) and the ARR-22 subset of NLPeer (Dycke et al., 2023), which include revisions of scientific papers and associated reviews. We extend the Re3-Sci dataset by annotating the remaining documents from the two source corpora totaling 1,780 scientific document revisions: 325 from NLPeer and 1,455 from F1000RD.

The automatic annotation consists of two steps: (1) **Revision Alignment (RA)** to identify sentence revision pairs as well as additions and deletions of sentences, and label them with action labels "Modify", "Add" or "Delete". We fine-tune a Llama2-13B classifier using *SeqC* achieving an accuracy of

99.3%, and employ a two-stage method as detailed in §D.1. (2). **EIC** to label the identified edits with intent labels. We use the best-performing Llama2-13B[5] classifier (§4), as it achieves the best performance, perfect AIR and high inference efficiency. A human evaluation of 10 randomly selected documents with 348 edits reveals 100% accuracy for RA and 90.5% accuracy for EIC (details in §D.2).

### 5.2 Basic Statistics and Subsets

The *Re3-Sci2.0* dataset includes 1,780 document revisions with 94,482 edits, each annotated with action and intent labels. The 325 documents from NLPeer are all from the NLP field (*nlp*), whereas the documents from F1000RD fall into three main subject domains: Natural Sciences (*nat*), Medical and Health Sciences (*med*) and Social Sciences (*soc*). Specific documents from the medical domain that provide brief reports on individual medical cases are separated from standard medical research papers to form a distinct *case* category. Similarly, documents from the natural sciences domain that provide technical reports on software or tools, primarily from computational biology, are separated into the *tool* category. §D.3 provides detailed definitions of the research domains and document categories, Table 4 presents statistics for each subset.

| | doc. | edit | d_word | d_sent. | d_edit |
|---|---|---|---|---|---|
| all | 1,780 | 94,482 | 4,650 | 201 | 53 |
| nlp | 325 | 29,782 | 5,775 | 262 | 92 |
| case (med) | 112 | 2,248 | 2,118 | 100 | 20 |
| med | 208 | 7,521 | 4,616 | 193 | 36 |
| tool (nat) | 162 | 7,143 | 3,505 | 170 | 44 |
| nat | 349 | 18,834 | 5,001 | 210 | 54 |
| soc | 46 | 2,466 | 4,888 | 206 | 54 |

Table 4: *Re3-Sci2.0* statistics and subsets. Presented are counts of documents and total sentence edits, and average counts of words, sentences and edits per document.

### 5.3 Analysis of Editing Behavior

As a resource, *Re3-Sci2.0* enables new empirical insights into the text editing behavior in the academic domain. We illustrate this analysis by investigating the following research questions:

**RQ1: How do successful revisions enhance scientific quality compared to unsuccessful ones?** We interpret increased review scores between document versions as indicators of successful revisions and improvements in scientific quality (more details in §E.1). We investigate the focus of authors' revisions by analyzing the document-based proportions of edit action and intent combinations as key variables. A value of 1 is assigned to successfully revised documents with increased review scores and 0 to unsuccessful ones. We then fit a binary logistic regression model to predict revision success, which is statistically significant with an LLR p-value of 0.001. Table 5 shows that focusing on modifications to enhance clarity and claims, and additions of new facts or evidence, significantly and positively influences the success of revisions. Additionally, Table 10 in §E.1 indicates that successful revisions include significantly more edits compared to unsuccessful ones.

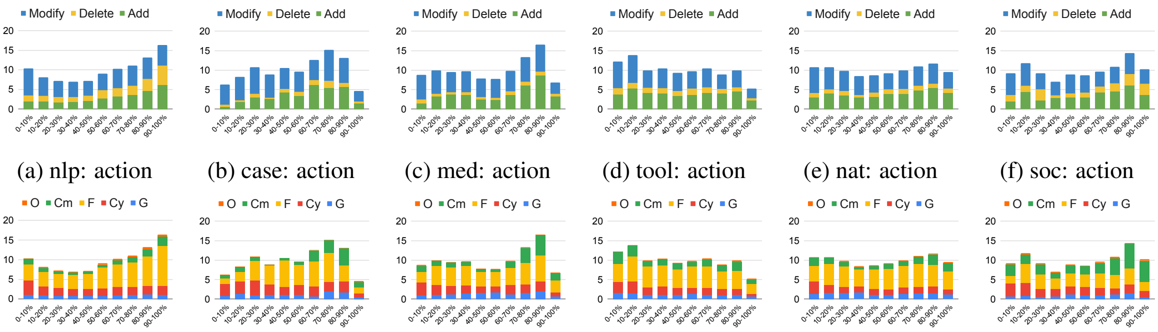
| | *coef* | *p-value* |
|---|---|---|
| Add, Fact/Evidence | **0.9341** | 0.003 |
| Add, Claim | 0.6116 | 0.221 |
| Delete, Fact/Evidence | 2.0920 | 0.061 |
| Delete, Claim | 2.9626 | 0.076 |
| Modify, Grammar | -0.5324 | 0.161 |
| Modify, Clarity | **1.0723** | 0.004 |
| Modify, Fact/Evidence | 0.3506 | 0.347 |
| Modify, Claim | **3.3392** | 0.040 |

Table 5: Results of the binary logistic regression. Presented are the regression coefficients for the variables. Bold values indicate statistical significance ($p < 0.05$).

**RQ2: How do human editing behaviors differ across various research domains and document categories?** To analyze human editing behaviors, we examine the proportions of action and intent combinations to reflect authors' editing focus (Figure 4) and analyze the distribution of edits across documents to identify editing location (Figure 3). A Kullback–Leibler Divergence (KL) analysis of the distributions across research domains and document categories is shown in Figure 7 in §E.2.

**Analysis indicates that human editing behaviors are consistent within the same research domain, despite variations in document categories.** For example, consider the *case* and *med* categories, both from the medical domain. Table 4 here and Figure 6 in §E.2 show that medical case reports (*case*) are generally shorter with fewer edits compared to other documents in the medical sciences (*med*). However, the revision focus of the authors appears similar, as illustrated in Figure 4b and Figure 4c. This similarity is further substantiated by the low KL values between *case* and *med* shown in Figure 7c in §E.2. The revision locations for both action and intent in *case* and *med* are also similar, as evidenced by comparing Figure 3b and Figure 3c, as well as Figure 3h and Figure 3i. These sim-

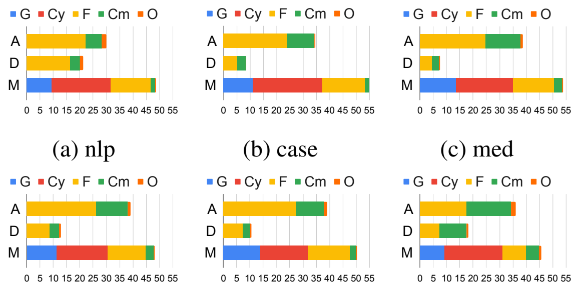
[5] We did not use the Llama3 classifiers since Llama3 was released after our auto-annotation process was completed.




(a) nlp: action (b) case: action (c) med: action (d) tool: action (e) nat: action (f) soc: action

(g) nlp: intent (h) case: intent (i) med: intent (j) tool: intent (k) nat: intent (l) soc: intent

Figure 3: Edit action and intent labels distribution over documents. The x-axis represents the relative sentence positions within documents. G: Grammar, Cy: Clarity, F: Fact/Evidence, Cm: Claim, O: Other.




(a) nlp (b) case (c) med

(d) tool (e) nat (f) soc

Figure 4: Combinations of edit action and intent labels across various categories. A: Add, D: Delete, M: Modify, G: Grammar, Cy: Clarity, F: Fact/Evidence, Cm: Claim, O: Other.

ilarities are supported by low KL scores between *case* and *med* in both Figure 7a and Figure 7b. Similarly, when comparing *tool* and *nat* across Figures 3, 4 and 7, it is evident that human editing focus and location are consistent within the natural sciences, regardless of different document categories.

Regarding **editing focus**, Figure 4 indicates that authors in the medical domain (*case* and *med*) and natural sciences (*tool* and *nat*) tend to make fewer deletions. In contrast, authors in NLP (*nlp*) and social sciences (*soc*) make more deletions, with the former emphasizing Fact/Evidence and the latter focusing more on Claim. Figure 7c in §E.2 further shows that the social sciences domain differs most substantially from other domains in terms of editing focus, as indicated by the high KL scores between *soc* and other domains. Regarding **editing location**, Figure 3 illustrates that in NLP, the final parts of documents are most frequently revised, primarily through additions and deletions of Fact/Evidence and Claim. In medical sciences (*case* and *med*), the 70-90% range of relative document positions is intensively revised, characterized by more additions and claim changes compared to other locations. In natural sciences (*tool* and *nat*) and social sciences (*soc*), edits tend to be more evenly distributed.

## 6 Conclusion

We have introduced a general framework for fine-tuning LLM classifiers, including four approaches, various LLM families, and training strategies. Experiments on EIC have demonstrated the strong encoding capabilities of LLMs. Our findings suggest that LLMs can be effectively fine-tuned as intent classifiers, outperforming fully fine-tuned PLMs and most advanced larger LLMs with instruction tuning. Among the approaches, the encoding-based *SeqC* approach has shown superiority in model performance, inference efficiency, and answer inclusion, while the cross network (*XNet*) also performs strongly. Using the best model achieving a macro average F1 score of 84.3, we have annotated a large-scale dataset of scientific document revisions, enabling in-depth empirical analysis of revision success and human editing behavior across various research domains. Our illustratory analysis suggests that (1) focus on Clarity and Claim modifications and Fact/Evidence additions significantly and positively impacts revisions success; (2) human editing focus and location remain consistent within the same research domain regardless of document categories but vary substantially across different domains. Our work paves the way for systematic investigation of LLMs for classification tasks and beyond. The general experimental framework is applicable to a wide range of classification tasks. The annotated dataset provides a robust foundation for multifaceted science-of-science research. The annotation models and processes can be applied to other domains as relevant data becomes available.

## Limitations

This study has several limitations that should be considered when interpreting the results. From a task and modeling perspective, this work focuses on edit intent classification, aiming to address this complex, challenging, yet underexplored task and facilitate crucial but understudied real-world applications for science-of-science analysis. The experimental results and discussions may not directly apply to other classification tasks. However, the proposed approaches and training strategies can be readily adapted to a wide range of classification tasks using our experimental framework, which we leave for future work.

From a data and analysis standpoint, the study's focus on English-language scientific publications stems from the limited availability of openly licensed scholarly publications in other languages. The use of Re3-Sci is driven by the need for high-quality and sufficiently large datasets for fine-tuning. Exploring the transferability of our findings to new languages, domains, and editorial workflows represents a promising direction for future research. When new data becomes available, our publicly available models can be used for annotation and analysis. Additionally, our experimental framework facilitates easy fine-tuning on other datasets and allows for systematic comparisons of various approaches and training strategies.

Finally, we highlight that our analysis serves an illustrative purpose. Its primary goal is to inspire researchers from other related disciplines to utilize natural language processing-based analysis in answering new questions about research work and scientific publishing. Enabled by the new dataset and methods, we leave the in-depth investigation of human editing behavior across research communities for future research.

## Ethics Statement

Re3-Sci and 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 dataset is available under a CC-BY-NC 4.0 license. The automatic annotation and analysis process does not involve the collection of any personal or sensitive information. For privacy protection, author metadata has been omitted from the data release.

## References


Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *ArXiv*, cs.LG/2305.14314.

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.

Santo Fortunato, Carl T. Bergstrom, Katy Börner, James A. Evans, Dirk Helbing, Staša Milojević, Alexander M. Petersen, Filippo Radicchi, Roberta Sinatra, Brian Uzzi, Alessandro Vespignani, Ludo Waltman, Dashun Wang, and Albert-László Barabási. 2018. Science of science. *Science*, 359(6379):eaao0185.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, cs.CL/2310.06825.

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.

Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Yejin Kim, Minkyung Cho, Jy yong Sohn, and Chanyeol Choi. 2024. Linq-embed-mistral:elevating text retrieval with improved gpt data through task-specific control and quality refinement. *Linq AI Research Blog*.

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.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *ArXiv*, cs.CL/2405.17428.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

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

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Sfr-embedding-mistral:enhance text retrieval with transfer learning. *Salesforce AI Research Blog*.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. 2023. In-context learning for text classification with many labels. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pages 173–184, Singapore. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Soham Parikh, Mitul Tiwari, Prashil Tumbade, and Quaizar Vohra. 2023. Exploring zero and few-shot techniques for intent classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 744–751, Toronto, Canada. Association for Computational Linguistics.

Parth Patwa, Simone Filice, Zhiyu Chen, Giuseppe Castellucci, Oleg Rokhlenko, and Shervin Malmasi. 2024. Enhancing low-resource llms classification with peft and synthetic data. *ArXiv*, cs.CL/2404.02422.

Youri Peskine, Damir Korenčić, Ivan Grubisic, Paolo Papotti, Raphael Troncy, and Paolo Rosso. 2023. Definitions matter: Guiding GPT for multi-label classification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4054–4063, Singapore. Association for Computational Linguistics.

Minh-Quang Pham, Sathish Indurthi, Shamil Chollampatt, and Marco Turchi. 2023. Select, prompt, filter: Distilling large language models for summarizing conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12257–12265, Singapore. Association for Computational Linguistics.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a general-purpose natural language processing task solver? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1339–1384, Singapore. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Hamidreza Rouzegar and Masoud Makrehchi. 2024. Enhancing text classification through LLM-driven active learning and human annotation. In *Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII)*, pages 98–111, St. Julians, Malta. Association for Computational Linguistics.

Qian Ruan, Ilia Kuznetsov, and Iryna Gurevych. 2024. Re3: A holistic framework and dataset for modeling collaborative document revision. *ArXiv*, cs.CL/2406.00197.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005, Singapore. 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. *ArXiv*, abs/2307.09288.

Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, and Ioannis Vlahavas. 2011. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12(71):2411–2414.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Yiming Wang, Zhuosheng Zhang, and Rui Wang. 2023. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8640–8665, Toronto, Canada. 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.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. 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.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Merging generated and retrieved knowledge for open-domain QA. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4710–4728, Singapore. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

# A Framework

**Input Tuning.** Table 7 provides examples of input texts in various settings, see §3.2 for details on input tuning.

**Language Models.** We select the LLMs based on four criteria: (1) they should be open-sourced to ensure reproducibility; (2) they should have a reasonable size to allow fine-tuning with moderate computing resources, while still varying in size (ranging from 6B to 13B) to assess the impact of model size; (3) there should be both instruction-fine-tuned and non-instruction-fine-tuned versions to study their performance differences and evaluate the effectiveness of instruction fine-tuning for different approaches (see RQ2 in §4.2), and (4) they should be recent and proven to be state-of-the-art or advanced on extensive NLP benchmarks (Zellers et al., 2019; Lin et al., 2022; Muennighoff et al., 2023). For the generation-based approach, we select an encoder-decoder PLM specifically designed for text-to-text generation to align with the approach's design. For the encoding-based approach, we use an encoder-only transformer model to assess its encoding capabilities in comparison to LLMs. Table 8 compares the models' features, including parameter size, number of layers, model dimension and architecture.

| base LM | r | a | d | acc. | m.f1 | AIR |
|---|---|---|---|---|---|---|
| **(a). Text generation** | | | | | | |
| Llama2-13B-Chat | 16 | 16 | 0.1 | 81.5 | 80.7 | 100 |
| | 128 | 16 | 0.1 | 81.8 | 81.1 | 100 |
| | 128 | 128 | 0.1 | 82.4 | 81.5 | 100 |
| | 256 | 16 | 0.1 | 80.8 | 80.7 | 100 |
| | 256 | 128 | 0.1 | 83.1 | 68.2 | 99.9 |
| | 256 | 256 | 0.1 | 83.6 | **82.8** | 100 |
| | 256 | 512 | 0.1 | 79.5 | 66.1 | 94.1 |
| | 512 | 16 | 0.1 | 81.7 | 66.9 | 99.9 |
| | 512 | 512 | 0.1 | 82.3 | 67.4 | 99.8 |
| | 1024 | 16 | 0.1 | 81.5 | 80.3 | 100 |
| | 1024 | 512 | 0.1 | 74 | 56.3 | 87.7 |
| | 1024 | 1024 | 0.1 | **84.5** | 68.9 | 99.9 |
| | 2048 | 16 | 0.1 | 81.7 | 67.1 | 99.9 |
| | 2048 | 2048 | 0.1 | 82 | 80.7 | 100 |
| **(b). Sequence Classification** | | | | | | |
| Llama2-7B-Chat | 16 | 16 | 0.1 | 83.9 | 82.2 | 100 |
| | 64 | 64 | 0.1 | 83.7 | 82.3 | 100 |
| | 128 | 128 | 0.1 | **84.4** | **82.8** | 100 |
| | 128 | 128 | 0.2 | 84.1 | 82.5 | 100 |
| | 256 | 256 | 0.1 | 83.8 | 82.0 | 100 |
| | 512 | 512 | 0.1 | 81.7 | 80.5 | 100 |

Table 6: Hyperparameters tuning. r: LoRA rank, a: LORA alpha, d: dropout. acc.: accuracy, m.f1: marco F1 score, AIR: Answer Inclusion Rate.

| | |
|---|---|
| **(a) Gen** | 1 *inst + natural input* |
| | Instruction: Classify the intent of the following sentence edit. The possible labels are: Grammar, Clarity, Fact/Evidence, Claim, Other.<br>INPUT:<br>OLD: The model is trained in a NVIDIA GeForce RTX 2080Ti GPU.<br>NEW: The model is trained in an NVIDIA GeForce RTX 2080Ti GPU.<br>RESPONSE: |
| | 2 *inst + structured input* |
| | <instruction><br>Classify the intent of the following sentence edit. The possible labels are: Grammar, Clarity, Fact/Evidence, Claim, Other.<br></instruction><br><input><br><old> The model is trained in a NVIDIA GeForce RTX 2080Ti GPU. </old><br><new> The model is trained in an NVIDIA GeForce RTX 2080Ti GPU. </new><br></input><br><response> |
| **(b) SeqC** | 1 *natural input* |
| | The model is trained in a NVIDIA GeForce RTX 2080Ti GPU.<br>The model is trained in an NVIDIA GeForce RTX 2080Ti GPU. |
| | 2 *structured input* |
| | <old> The model is trained in a NVIDIA GeForce RTX 2080Ti GPU. </old><br><new> The model is trained in an NVIDIA GeForce RTX 2080Ti GPU. </new> |
| | 3 *inst + structured input* |
| | Classify the intent of the following sentence edit. The possible labels are: Grammar, Clarity, Fact/Evidence, Claim, Other.<br><old> The model is trained in a NVIDIA GeForce RTX 2080Ti GPU. </old><br><new> The model is trained in an NVIDIA GeForce RTX 2080Ti GPU. </new> |

Table 7: Examples of different input types.

## B Experimental Details

We fine-tune all linear layers of the LLMs using QLoRA (Dettmers et al., 2023), tuning parameters such as LoRA rank ($r$), LoRA alpha ($a$), and dropout ($d$) during initial experiments. Based on the results in Table 6, we set the parameters as follows: for approach *Gen*, we set $r$=256, $a$= 256, $d$=0.1; for approaches *SeqC*, *SNet*, and *XNet*, the settings are $r$=128, $a$=128, $d$=0.1. The small PLMs, T5 and RoBERTa, are fully fine-tuned with all weights being directly updated.

For approach *Gen*, the output token limit is set to ten. We define the metric Answer Inclusion Rate (AIR) as the percentage of samples where a label string falls within the ten output tokens regardless of correctness. If the output tokens do not contain any label string, the prediction is considered a failure. When using RoBERTa for approach *SeqC*, the the first token representation is used as the input for classification.

For all approaches and base LMs, the models are fine-tuned for ten epochs on the training set, with checkpoints saved after each epoch. The final model selection is determined based on evaluation results from the validation set, and its performance is subsequently assessed on the test set. For approaches *SeqC*, *SNet*, and *XNet*, a single NVIDIA A100 or H100 GPU with 80GB memory is utilized. Approach *Gen* requires two such GPUs.

In Table 2, the human performance is calculated from individual human annotations in Re3-Sci and the gold labels aggregated by majority voting. For the GPT-4 baselines, the gpt-4-turbo model released in April 2024 was used. GPT-4 (ICL+CoT) uses the default ICL examples and CoT formats provided by Ruan et al. (2024). In Table 3, the structured input format (§3.2) without task instructions is used.

## C Discussion

Figure 5 compares the three metrics for the four approaches using Llama2-13B as the base LM. Approach *SeqC* achieves perfect AIR, the best performance, and a 12x inference speedup compared to approach *Gen* and a 4x speedup compared to approaches *SNet* and *XNet*.

| models | size | #layers | dim | inst | architecture |
|---|---|---|---|---|---|
| GPT-j (2021) | 6B | 28 | 4096 | no | decoder-only |
| Mistral-Instruct (2023) | 7B | 32 | 4096 | yes | decoder-only |
| Llama2-7B (2023) | 7B | 32 | 4096 | no | decoder-only |
| Llama2-7B-Chat (2023) | 7B | 32 | 4096 | yes | decoder-only |
| Llama2-13B (2023) | 13B | 40 | 5120 | no | decoder-only |
| Llama2-13B-Chat (2023) | 13B | 40 | 5120 | yes | decoder-only |
| Llama3-8B (2024) | 8B | 32 | 4096 | no | decoder-only |
| Llama2-8B-Chat (2024) | 8B | 32 | 4096 | yes | decoder-only |
| RoBERTa-base (2019) | 125M | 12 | 768 | no | encoder-only |
| T5-base (2020) | 220M | 12 | 768 | no | encoder-decoder |

Table 8: Language model comparisons. Presented are the parameter size, number of layers, model dimension, whether the model is fine-tuned for instruction-following, and the transformer architecture of each model.

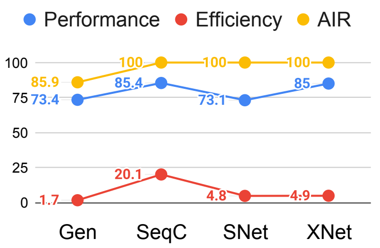



Figure 5: Approaches comparison. AIR: Answer Inclusion Rate, performance: accuracy, efficiency: the number of samples processed per second during inference.

# D Auto-annotation

## D.1 Revision Alignment

Both source datasets, F1000RD and NLPeer contain structured documents organized into sections and paragraphs, which we refine to sentences using the method proposed by Ruan et al. (2024). To manage the extensive comparison scope resulting from candidate pairs within long document revisions, we employ a two-stage approach for revision alignment. Initially, we utilize the lightweight pre-alignment algorithm proposed by Ruan et al. (2024), which efficiently identifies candidates and accurately extracts revision pairs with a precision of 0.99, while maintaining minimal computational cost. However, the recall for alignment (0.92) is relatively low due to the algorithm's stringent aligning rules. To address this, we fine-tune a Llama2-13B model using approach *SeqC* with instruction and structured input on the revision alignment data from Re3-Sci. This achieves a precision of 0.99 for non-alignment and a recall of 0.99 for alignment, perfectly enhancing the pre-alignment algorithm. We selectively apply the fine-tuned model to non-aligned candidates identified by the pre-alignment algorithm. This approach allows us to identify missing revision pairs without significantly increasing computational overhead. The identified revision pairs are annotated with the action label "Modify". Sentences in the new document that do not align with any in the old document are labeled as "Add", while unmatched sentences in the old document are marked as "Delete".

## D.2 Human Evaluation

A human evaluation of the labeled *Re3-Sci2.0* data is conducted, randomly selecting 10 documents with 348 edits. The evaluation reveals 100% accuracy for revision alignment, and for edit intent classification, a 90.5% accuracy and a macro average F1 score of 86.4. Table 9 indicates that the failures in edit intent classification are particularly associated with the low-resource "Other" class in the training set (Ruan et al., 2024), while the other classes have substantial F1 scores.

## D.3 Subject Domains and Document Categories

The F1000RD documents fall into three main subject domains according to the F1000RD website[6]:

- Medical and health sciences focuses on the provision of healthcare, the prevention and treatment of human diseases and interventions and technology for use in healthcare to improve the treatment of patients.

- Natural sciences comprises the branches of science which aim to describe and understand the fundamental processes and phenomena that define our natural world, including both life sciences and physical sciences.

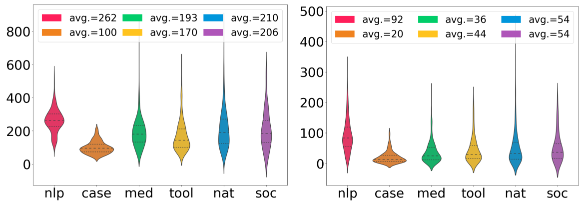
[6] https://f1000research.com/

| class | Total | | Grammar | | | Clarity | | | Fact/Evidence | | | Claim | | | Other | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 348 | | 17 | | | 61 | | | 158 | | | 88 | | | 24 | | |
| metrics | Acc. | M. F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| | 90.5 | 86.4 | 73.9 | 100 | 85 | 84.1 | 95.1 | 89.2 | 92.6 | 94.9 | 93.8 | 97.4 | 85.2 | 90.9 | 88.2 | 62.5 | 73.2 |

Table 9: Human evaluation of edit intent classification. Displayed are the overall accuracy (Acc.), macro average F1 score (M. F1), and precision (P), recall (R), and F1 score for each label. The failures are particularly associated with the low-resource "Other" class in the training set (Ruan et al., 2024), while the other classes have substantial F1 scores.

| | *successful* | *unsuccessful* |
|---|---|---|
| #Grammar | 5.5 | 6.1 |
| #Clarity | **9.3** | **7.3** |
| #Fact/Evidence | 22.0 | 19.1 |
| #Claim | **8.6** | **5.9** |
| #Other | 1.0 | 0.7 |
| #edits | **46.4** | **39.1** |

Table 10: Average number of edits per intent per document and average number of total edits per document. Values are bolded if two-sample t-tests indicate a significant difference between the successful and unsuccessful groups, with p<0.05.




(a) doc length (s) (b) doc edits (s)

Figure 6: Comparison of categories by (a) document sentence count and (b) sentence edits within documents.

- Social sciences subject areas seeks to understand social relationships, societal issues and the ways in which people behave and shape our world.

The six document categories are defined as:

- *nlp*: documents from the NLPeer corpus that present research on Natural Language Processing
- *case (med)*: specific F1000RD documents from the medical and health sciences that provide short reports on individual medical cases
- *med*: other research papers from the medical and health sciences domain within the F1000RD dataset
- *tool (nat)*: specific F1000RD documents from the natural sciences domain that provide technical reports on software or tools, primarily from computational biology
- *nat*: other research papers from the natural sciences field within the F1000RD dataset
- *soc*: documents from the social sciences domain within the F1000RD dataset

Documents that do not fit into any domains or belong to more than one domain are excluded from the divisions.

# E Edit Analysis

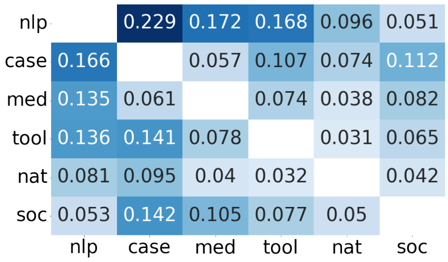
## E.1 Successful vs. Unsuccessful Revisions

We interpret increased reviewer scores as indicators of successful revisions and improvements in scientific quality. Reviewers in the F1000RD community evaluate publications using one of three decisions: "reject," "approve-with-reservations," or "approve", which we convert into numeric values.[7] Document revisions that result in an increased average reviewer score are considered successful, while those that do not are deemed unsuccessful. Among the 849 F1000RD documents with reviewer scores for both initial and final versions, 575 are categorized as successful and 274 as unsuccessful. Documents from the NLPeer corpus lack final reviewer scores for their final versions; however, since all are accepted to a venue, we assume that the 325 documents have all undergone successful revisions. Given that our objective for RQ1 in §5.3 is to compare successful revisions with unsuccessful ones, we utilize the categorized F1000RD documents for the analysis, as the NLPeer documents lack unsuccessful samples.

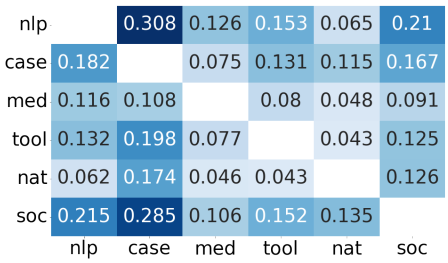
Table 10 shows that successful revisions contain significantly more edits than unsuccessful ones, particularly with more changes in Clarity and Claim.

## E.2 Editing Behavior across Research Domains and Document Categories

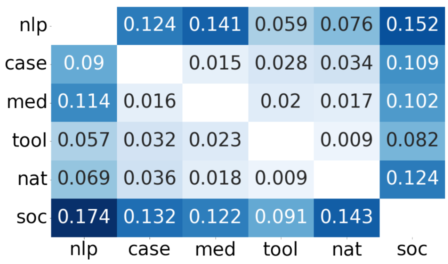
[7]"reject":1, "approve-with-reservations":2, "approve":3




(a) action location




(b) intent location




(c) label combination

Figure 7: Kullback–Leibler (KL) Divergence analysis of the distributions across categories for (a) action location (Figure 3, 1st line) (b) intent location (Figure 3, 2nd line) and (c) edit action and intent combinations (Figure 4). The higher the KL divergence, the greater the difference between the distributions.