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

Qian Ruan, Ilia Kuznetsov, Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP Lab)
Department of Computer Science and Hessian Center for AI (hessian.AI)
Technical University of Darmstadt
www.ukp.tu-darmstadt.de

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. We investigate the generalizability of these findings on five further classification tasks. To demonstrate the proposed methods and address the data shortage for empirical edit analysis, we use our best-performing EIC model to create *Re3-Sci2.0*, a new large-scale dataset of 1,780 scientific document revisions with over 94k labeled edits. The quality of the dataset is assessed through human evaluation. The new dataset enables an in-depth empirical study of human editing behavior in academic writing. We make our experimental framework[1], models and data[2] publicly available.

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.

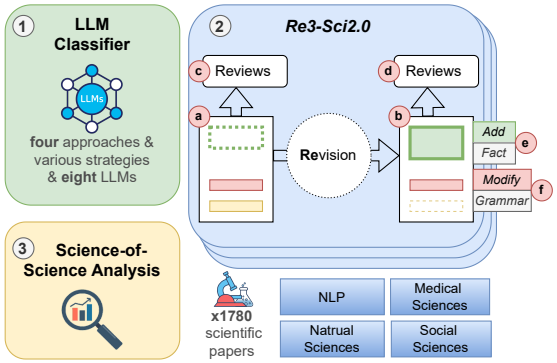


Figure 1: In this work, we (1). present a general framework to explore the classification capabilities of LLMs, conducting extensive experiments and systematic comparisons on the EIC task; (2). use the best model to create the *Re3-Sci2.0* dataset, which comprises 1,780 scientific document revisions (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

[1] https://github.com/UKPLab/llm_classifier
[2] https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/4355

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 15049–15067
November 12-16, 2024

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 inputs. 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.[3]

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 LLMs fine-tuned with encoding-based approaches demonstrate superior classification capabilities for EIC, achieving state-of-the-art (SOTA) performance. To demonstrate the versatility of our framework, we apply it to five further classification tasks and investigate the generalizability of our insights (§4). To illustrate the application of the proposed methods for EIC and address the lack of data for extensive edit analysis, we use our best-performing 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, base PLMs and LLMs,[4] supplemented by evaluation on five further classification tasks.
- A large dataset of 1,780 scientific document revisions with 94,482 edits, annotated by our best EIC 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 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.

[3]Note that direct performance comparison in Table 1 is not possible due to different datasets, label sets and data sizes, but they illustrate the inherent difficulty of EIC despite data variations.

[4]While current LLM terminology requires further precision, as discussed in Rogers and Luccioni (2024), we use the terms "LLMs" and "PLMs" for readability. "LLMs" refers to latest-generation large-scale language models, such as Mistral-Instruct, LLaMA, and GPT-4, which cannot be trained or fully fine-tuned on one or two modern GPUs. In contrast, "PLMs" denotes earlier smaller pre-trained language models, such as T5, RoBERTa, and other BERT variants, which can be trained and fully fine-tuned using one or two GPUs. Details on the language models are provided in §3.4

While existing works (Zhang et al., 2016; Yang et al., 2017; Kashefi et al., 2022; Du et al., 2022; Jiang et al., 2022; Ruan et al., 2024) demonstrate the critical role of intent labels in understanding human editing, they also highlight the challenges of manually labeling edit intent, which requires expert annotators, specialized annotation tools, and extensive training (Jiang et al., 2022; Ruan et al., 2024). The high costs and efforts of manual labeling limit the size of available datasets (Table 1), restrict large-scale studies of human editing behavior and motivate the need for effective NLP automation in EIC and the creation of larger labeled datasets.

From the modeling perspective, several works (Zhang et al., 2016; Yang et al., 2017; Kashefi et al., 2022) have primarily investigated automatic EIC using 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 NLP task (§3). To address the shortage of revision datasets, we use our most efficient and high-performing EIC model to create the new, large-scale Re3-Sci2.0 dataset. We provide a comprehensive overview of the entire pipeline — model selection (§4), annotation (§5), and revision analysis (§5.3) — to ensure a complete and reproducible process for generating high-quality, large-scale automatically labeled revision datasets with LLMs.

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[5] (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 our work focuses on the challenging and crucial EIC task (§1), our framework is applicable to a wide range of classification tasks, as demonstrated by our experiments with additional tasks in §4.3.

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-class pair 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

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

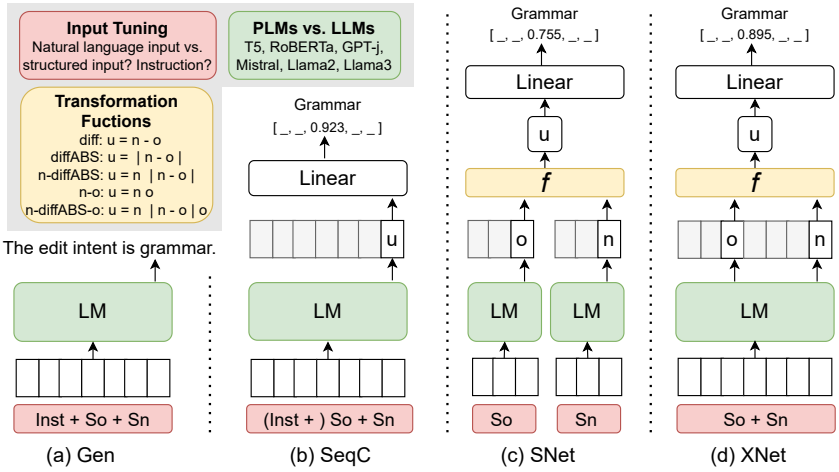


Figure 2: Proposed approaches with a systematic investigation of the key components: input types (red), language models (green), and transformation functions (yellow). See §3 and §4 for details.

into a k-dimensional logit vector, where the maximum value indicates the predicted label.

- **Approach *SNet*** employs a Siamese architecture akin to SBERT (Reimers and Gurevych, 2019) 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) converts these into a single representation u for classification.
- **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 transformed into a single representation u by a 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 8 in §A.

3.3 Transformation Functions

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

$$f_{diff} : u = n - o \quad (1)$$

$$f_{diffABS} : u = |n - o| \quad (2)$$

$$f_{n-diffABS} : u = n \oplus |n - o| \quad (3)$$

$$f_{n-o} : u = n \oplus o \quad (4)$$

$$f_{n-diffABS-o} : u = n \oplus |n - o| \oplus o \quad (5)$$

where $\oplus$ represents vector concatenation, - denotes vector subtraction, and | | indicates that absolute values are taken from the subtraction. The intuition is to incorporate the differences between sentence embeddings in the transformation process, as the EIC task relies on analyzing the variations between two versions of a text. The five proposed 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

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†	69†	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	1.8	4.8	79.9	78.1	100	78.3 (↓1.6)	78.0 (↓0.1)	100
GPT-j	6B	12.6	11.2	68.9	32.8	21.0	97.6	21.2 (↓11.6)	15.4 (↓5.6)	86.8 (↓10.8)
Mistral-Instruct	7B	28.0	24.0	99.9	68.5	63.4	100	62.8 (↓5.7)	59.2 (↓4.2)	100
Llama2-7B	7B	21.4	12.2	78.2	34.3	24.7	100	60.4 (↑26.1)	47.6 (↑22.9)	88.7 (↓11.3)
Llama2-7B-Chat	7B	12.1	8.6	85.2	63.0	49.2	100	72.4 (↑9.4)	55.0 (↑5.8)	88.5 (↓11.5)
Llama2-13B	13B	13.8	5.2	93.3	50.9	39.5	99.9	73.4 (↑22.5)	67.6 (↑28.1)	85.9 (↓14.0)
Llama2-13B-Chat	13B	0.5	1.9	2.0	75.5	72.9	100	83.6 (↑8.1)	82.8 (↑9.9)	100
Llama3-8B	8B	14.0	13.3	77.8	79.4	79.1	95.4	83.3 (↑3.9)	82.1 (↑3.0)	99.9 (↑4.5)
Llama3-8B-Instruct	8B	12.6	17.3	47.3	84.1†	82.4†	100	**84.7**† (↑0.6)	**83.7**† (↑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	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† (↑0.5)	**84.3**† (↑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	13.0	84.1	82.3†	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†	82.2	**85.6**† (↑1.2)	**84.3**† (↑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 † denotes the best-performing LM within each setting. The best metrics from each approach are highlighted in bold.

Llama2-13B-Chat (Touvron et al., 2023), Llama3-8B and Llama3-8B-Instruct[6], and compare them with two 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-class 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

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

	(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†
Llama2-7B-Chat	60.7	56.5	72.4	71.4	65.4	64.7	58.7	55.3	68.5†	67.6
Llama2-13B	62.4	59.3	73.1	72.4	67.5	67.2	61.0†	59.1†	66.0	67.2
Llama2-13B-Chat	63.7†	61.6†	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†	68.7†	58.6	56.6	64.8	63.8
Llama3-8B-Instruct	59.9	56.6	**73.3**†	**72.9**†	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**†	84.4	83.4	84.6†	83.7†
Llama2-13B-Chat	84.3	82.9	**85.2**†	83.7†	84.5	83.6	84.9	83.7†	84.6†	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†	83.4†	84.5	83.2	85.1†	83.7	85.1†	83.7†	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, † denotes the best-performing LM within each setting. The best metrics from each approach are in bold.

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, and achieving new state-of-the-art (SOTA) performance on the Re3-Sci dataset. 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 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 substantially surpasses the previous SOTA results achieved by 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*, most of the eight LLMs outperform a fully fine-tuned RoBERTa across various input formats, highlighting the superior encoding capabilities of LLMs. Table 2(a) shows that using approach *Gen* with structured inputs, 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 fine-tuned models, marked with † in Tables 2 and 3, reveals that the 13B Llama2 and 8B Llama3 models demonstrate the greatest potential and achieve the best results. Additionally, we observe that using the *Gen* approach (Table 2(a)), instruction-fine-tuned LLMs consistently outperform their non-instruction-fine-tuned counterparts, with statistical significance supported by paired one-sided two-sample t-tests (Student, 1908) and one-sided Wilcoxon signed-rank tests (Wilcoxon, 1945). This performance improvement is likely attributable to the enhanced capability of instruction-fine-tuned models to comprehend complex task instructions and label tags. However, in *SeqC*, *SNet* and *XNet* approaches, there are no consistent performance differences between the chat and non-chat versions of LLMs.

RQ3: Which approach is most effective? Overall, the *SeqC* approach demonstrates superior performance, answer inclusion rate (AIR), and inference efficiency. Table 2(a) indicates that generative models encounter AIR issues even after fine-tuning, suggesting that the generation-based approach is not optimal in practice due to its lack of robustness and difficulty in control. In terms of performance, approaches *SeqC* and *XNet* are superior. The cross network (*XNet*) consistently and substantially outperforms the Siamese network (*SNet*) when using the same LLMs and transformation functions (Table 3). The *SeqC* approach demonstrates notable superiority in inference efficiency, measured by the number of samples processed per second during inference, making it particularly well-suited for application to large datasets. Figure 3 compares the four approaches across the three aspects, using Llama2-13B as the base language model. The

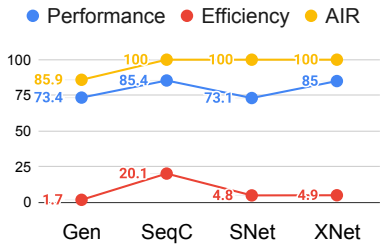


Figure 3: Approaches comparison using Llama2-13B as the base language model. AIR: Answer Inclusion Rate; performance: accuracy; efficiency: the number of samples processed per second during inference.

SeqC approach achieves perfect AIR, the best performance, and a 12x inference speedup compared to the *Gen* approach 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. For *XNet*, the best-performing functions are $f_{n-diffABS}$ and $f_{diffABS}$, as indicated in Table 3(d). These results align with our intuition that the EIC task focuses on analyzing variations between text versions, and incorporating sentence embedding differences proves to be effective.

4.3 Generalization Evaluation

We assess the generalization of our findings from the EIC task across five additional classification tasks from the MTEB benchmark (Muennighoff et al., 2023). The selected tasks comprise three binary pair classification tasks, in which the inputs are sentence pairs and the outputs are binary labels, along with two multi-class single-input classification tasks, where the inputs consist of individual sentences and the output labels are multi-class. The former group features a similar input architecture to that of EIC, allowing for the application of all four approaches. The latter type exhibits output complexity similar to that of EIC, featuring multiple potential labels.

Our evaluations across the five tasks provide compelling evidence that: (1) LLMs can be fine-tuned to operate as good classifiers, achieving SOTA results on the additional tasks; (2) among the eight tested LLMs, the 13B Llama2 and 8B Llama3 models exhibit the greatest potential and achieve best results; and (3) the encoding-based *SeqC* approach proves to be the most effective, demonstrating significantly superior performance, inference efficiency, and perfect AIR. Appendix §C gives details on the datasets and tasks, experimental settings and results, as well as further discussion.

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[7] 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 13 in §E.1 indicates that successful revisions include significantly more edits compared to unsuccessful ones.

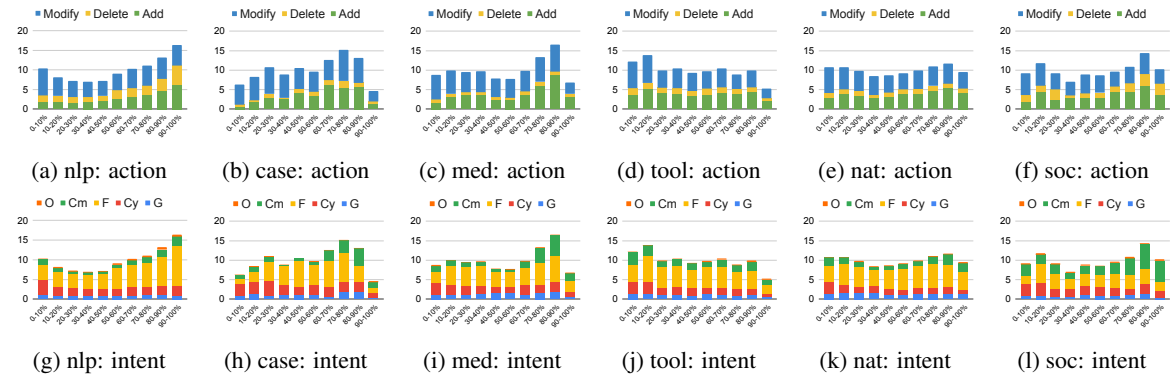
	coef	*p-value*
Add, Fact/Evidence	**0.9341**	0.003
Add, Claim	0.6116	0.221
Delete, Fact/Evidence	2.0920	0.061
Delete, Claim	2.9626	0.076
Modify, Grammar	-0.5324	0.161
Modify, Clarity	**1.0723**	0.004
Modify, Fact/Evidence	0.3506	0.347
Modify, Claim	**3.3392**	0.040

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 5) and analyze the distribution of edits across documents to identify editing location (Figure 4). A Kullback–Leibler Divergence (KL) analysis of the distributions across research domains and document categories is shown in Figure 6 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 shows 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 5b and Figure 5c. This similarity is further substantiated by the low KL values between *case* and *med* shown in Figure 6c in §E.2. The revision locations for both action and intent in *case* and *med* are also similar, as evidenced by comparing Figure 4b and Figure 4c, as well as Figure 4h and Figure 4i. These similarities are supported by low KL scores between *case* and *med* in both Figure 6a and Figure 6b. Similarly, when comparing *tool* and *nat* across Figures 4, 5 and 6, it is evident that

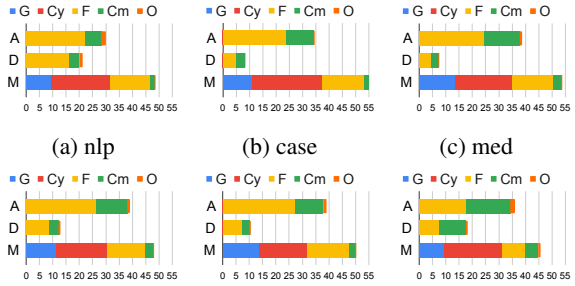
[7]We did not use the Llama3 classifiers since Llama3 was released after our auto-annotation process was completed.



(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 4: 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 5: 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.

human editing focus and location are consistent within the natural sciences, regardless of different document categories.

Regarding **editing focus**, Figure 5 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 6c 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 4 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. Extensive experiments on EIC have demonstrated that LLMs can be effectively fine-tuned as intent classifiers, outperforming fully fine-tuned PLMs and achieving SOTA results. Among the four approaches, the encoding-based *SeqC* approach has shown superiority in model performance, inference efficiency, and answer inclusion. Furthermore, we have demonstrated the versatility of our framework and evaluated the generalizability of our findings on five additional classification tasks.

Using the best-performing EIC model, 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 new dataset provides a robust foundation for multifaceted research in human editing in scientific domain and beyond. The annotation models and the labeling process are reusable and can be applied to generate new high-quality, large-scale automatically labeled revision datasets, as more raw 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. While we conducted extensive generalization evaluations, the experimental results and discussions may not be directly applicable to other classification tasks. However, the proposed approaches and training strategies can be readily adapted to other classification tasks within our experimental framework.

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 use natural language processing to explore new questions about editing, academic publishing and communication. 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.

Acknowledgements

This work is part of the InterText initiative[8] at the UKP Lab. This work has been funded by the German Research Foundation (DFG) as part of the PEER project (grant GU 798/28-1) and co-funded by the LOEWE Distinguished Chair "Ubiquitous Knowledge Processing", LOEWE initiative, Hesse, Germany (Grant Number: LOEWE/4a//519/05/00.002(0002)/81). It has been also co-funded by the European Union (ERC, InterText, 101054961). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

We gratefully acknowledge the support of Microsoft with a grant for access to OpenAI GPT models via the Azure cloud (Accelerate Foundation Model Academic Research). We thank Furkan Şahinuç, Serwar Basch and Tim Baumgärtner for their valuable feedback and suggestions on a draft of this paper. We would also like to express our gratitude to Kexin Wang, Nils Dycke, Jan Buchmann, Dennis Zyska, Serwar Basch and Falko Helm for their insightful discussions throughout the project.

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

[8] https://intertext.ukp-lab.de/

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.

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.

Anna Rogers and Alexandra Sasha Luccioni. 2024. Position: Key claims in llm research have a long tail of footnotes. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 42647–42665.

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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4635–4655, Bangkok, Thailand. Association for Computational Linguistics.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. Adversarial domain adaptation for duplicate question detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1056–1063, Brussels, Belgium. Association for Computational Linguistics.

Student. 1908. The probable error of a mean. *Biometrika*, 6(1):1–25.

Yu Su and Xifeng Yan. 2017. Cross-domain semantic parsing via paraphrasing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1235–1246, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. SemEval-2015 task 1: Paraphrase and semantic similarity in Twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 1–11, Denver, Colorado. 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 8 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 using QLoRA (Dettmers et al., 2023) 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).[9] We select the small pre-trained language models (PLMs) that can be fully fine-tuned with equivalent computing resources. For the generation-based approach, we select an encoder-decoder PLM (i.e., T5) 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 (i.e., RoBERTa) to assess its encoding capabilities in comparison to LLMs. Table 6 compares the models' features, including parameter size, number of layers, model dimension and architecture.

[9]As of April 2024

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 6: 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.

base LM	r	a	d	acc.	m.f1	AIR
(a). Gen						
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	81.9	99.9
	256	256	0.1	**83.6**	**82.8**	100
	256	512	0.1	79.5	79.3	94.1
	512	16	0.1	81.7	80.3	99.9
	512	512	0.1	82.3	80.9	99.8
(b). SeqC						
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 7: Hyperparameters tuning. r: LoRA rank, a: LORA alpha, d: dropout. acc.: accuracy, m.f1: marco F1 score, AIR: Answer Inclusion Rate.

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 7, 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

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

Table 8: Examples of different input types.

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 Generalization Evaluation

We assess the generalization of our findings from the EIC dataset across five additional classification tasks from the MTEB benchmark (Muennighoff et al., 2023).

C.1 Tasks and Datasets

The selected tasks comprise three binary pair classification tasks, in which the inputs are sentence pairs and the outputs are binary labels, along with two multi-class single-input classification tasks, where the inputs consist of individual sentences and the output labels are multi-class. The former group features a similar input architecture to that of EIC, allowing for the application of all four approaches. The latter type exhibits output complexity comparable to that of EIC, encompassing multiple potential labels. Below, we provide brief descriptions of each dataset as reported by Muennighoff et al. (2023).

Binary Pair Classification:

- **SprintDuplicateQuestions (SDQ)** (Shah et al., 2018). Collection of questions from the Sprint community. The goal is to classify a pair of sentences as duplicates or not.
- **TwitterSemEval2015 (TSE)** (Xu et al., 2015). Paraphrase-Pairs of Tweets from the SemEval 2015 workshop. The goal is to classify a pair of tweets as paraphrases or not.
- **TwitterURLCorpus (TUC)** (Su and Yan, 2017). Paraphrase-Pairs of Tweets. The goal

Task Type		Binary Pair Classification									Single Input Classification					
Dataset		SDQ			TSE			TUC			EC			TSEC		
Metric		acc.	m.f1	AIR	acc.	m.f1	AIR	acc.	m.f1	AIR	acc.	m.f1	AIR	acc.	m.f1	AIR
Previous SOTA		99.9	93.7		88.4	73.7		90.0	80.5		93.4	90.1		80.9	81.2	
T5	*Gen*	99.1	86.4	100	87.9	83.8	100	86.8	84.1	100	93.1	89.5	100	80.0	80.1	100
RoBERTa	*SeqC*	99.7	93.5	100	88.6	79.4	100	88.5	85.9	100	93.2	89.3	100	79.1	79.3	100
GPT-j	*Gen*	96.2	69.2	100	81.4	76.6	99.7	86.2	82.9	99.9	90.0	85.2	100	65.9	69.1	91.4
	SeqC	99.8	95.6	100	89.4	84.8	100	90.2	87.5	100	93.0	88.2	100	78.3	78.6	100
	SNet	98.7	49.7	100	78.1	43.8	100	73.9	42.5	100	-	-	-	-	-	-
	XNet	98.7	49.7	100	78.1	43.8	100	73.9	42.5	100	-	-	-	-	-	-
Mistral-Instruct	*Gen*	99.8	95.7	100	89.7	85.5	100	75.5	50.3	100	90.5	84.9	99.9	80.0†	80.3†	100
	SeqC	99.9	97.3	100	78.1	43.8	100	90.5	87.8	100	92.5	88.2	100	79.2	79.5	100
	SNet	98.2	79.1	100	68.5	55.5	100	83.0	76.5	100	-	-	-	-	-	-
	XNet	99.7	94.2	100	90.9$^{\#}$	86.6	100	87.2	84.6	100	-	-	-	-	-	-
Llama2-7B	*Gen*	99.5	92.1	100	89.5	84.5	100	90.1	87.6	100	79.8	71.6	100	73.0	73.9	98.7
	SeqC	99.6	93.4	100	89.6	86.0	100	90.0	87.8	100	93.5	89.2	100	77.8	78.2	100
	SNet	99.3	89.2	100	89.6	84.9	100	88.8$^{!}$	85.4	100	-	-	-	-	-	-
	XNet	99.7	94.2	100	90.8	87.2$^{\#}$	100	89.4	86.6	100	-	-	-	-	-	-
Llama2-7B-Chat	*Gen*	99.5	91.5	100	88.6	84.9	100	89.9	87.3	100	89.3	84.5	99.7	75.5	75.8	100
	SeqC	99.9	98.2	100	89.9	85.9	100	90.9	88.5	100	93.8	**89.6***	100	78.3	78.6	100
	SNet	99.5	91.7	100	87.1	81.3	100	88.4	84.0	100	-	-	-	-	-	-
	XNet	99.4	90.8	100	90.4	86.5	100	90.7	88.3	100	-	-	-	-	-	-
Llama2-13B	*Gen*	99.6	92.7	100	89.3	85.2	100	90.9	88.6†	100	90.0	84.8	99.9	76.3	76.8	100
	SeqC	99.7	94.9	100	**91.3***	**87.6***	100	90.9	88.2	100	93.2	88.8	100	79.0	79.4	100
	SNet	99.8	96.4	100	88.4	84.5	100	88.2	85.4	100	-	-	-	-	-	-
	XNet	99.9	98.2	100	89.4	83.5	100	90.1	86.5	100	-	-	-	-	-	-
Llama2-13B-Chat	*Gen*	99.9†	98.2†	100	89.9†	86.4†	100	91.0†	88.2	100	91.0†	86.9†	99.9	77.0	77.7	98.5
	SeqC	99.7	94.2	100	90.5	86.8	100	**91.3***	**88.9***	100	93.1	87.7	100	79.8	80.1	100
	SNet	99.6	92.5	100	90.4$^{!}$	85.7$^{!}$	100	88.7	85.5	100	-	-	-	-	-	-
	XNet	99.8	95.7	100	89.9	86.4	100	91.0$^{\#}$	88.6$^{\#}$	100	-	-	-	-	-	-
Llama3-8B	*Gen*	99.6	92.2	100	89.9†	86.4†	100	89.7	87.1	100	56.5	47.1	98.5	75.6	75.9	100
	SeqC	99.7	94.0	100	91.0	87.1	100	89.8	87.2	100	93.5	88.7	100	**80.5***	**80.8***	100
	SNet	99.2	87.5	100	89.0	84.4	100	86.7	80.5	100	-	-	-	-	-	-
	XNet	99.8	95.7	100	90.2	87.0	100	89.8	86.7	100	-	-	-	-	-	-
Llama3-8B-Instruct	*Gen*	99.3	88.9	100	89.5	86.1	100	88.6	84.3	100	56.7	47.7	99.7	77.4	77.8	100
	SeqC	**100***	**99.1***	100	90.6	87.0	100	89.8	87.4	100	**94.1***	**89.6***	100	78.6	78.9	100
	SNet	99.9$^{!}$	97.2$^{!}$	100	88.1	82.9	100	88.6	85.6$^{!}$	100	-	-	-	-	-	-
	XNet	**100**$^{\#}$	**99.1**$^{\#}$	100	89.6	86.3	100	90.4	88.0	100	-	-	-	-	-	-

Table 9: Results on the MTEB benchmark. Reported are accuracy (acc.), macro average F1 score (m. f1) and Answer Inclusion Rate (AIR) on the test set of each dataset. The best metrics for each dataset are in bold. The best-performing approach for each LM is underlined. †, *, $^{!}$ and $^{\#}$ denote the best-performing LM within each approach (Gen, SeqC, SNet and XNet) for each dataset.

is to classify a pair of tweets as paraphrases or not.

Multi-class Single-input Classification:

- **EmotionClassification (EC)** (Saravia et al., 2018). Dataset of English Twitter messages with six basic emotions: anger, fear, joy, love, sadness, and surprise.
- **TweetSentimentExtractionClassification (TSEC)**.[10] TweetSentimentExtraction Dataset from Kaggle competition. Sentiment classification of tweets as neutral, positive or negative.

[10] https://www.kaggle.com/competitions/tweet-sentiment-extraction

C.2 Experimental Details

The binary pair classification datasets are imbalanced, containing a substantial number of negative samples. To balance our training data, we retain positive samples from the original training set and randomly select an equivalent number of negative samples, resulting in 1,786 training samples for the SDQ dataset, 5,494 for TSE and 5,000 for the TUC dataset. For each dataset, we randomly select 1k validation samples and 2k test samples, ensuring that the original imbalanced label proportions are maintained. For the single-input classification datasets, we randomly select 16k training samples, along with 1k validation samples and 2k test samples, all preserving the original label proportions.

We fine-tune all linear layers of the LLMs using

Task Type	Binary Pair Class.						Single Input Class.			
Dataset	SDQ		TSE		TUC		EC		TSEC	
Metric	acc.	m.f1	acc.	m.f1	acc.	m.f1	acc.	m.f1	acc.	m.f1
RQ1: Are fine-tuned LLMs good classifiers?										
(1). The best results are achieved with the SeqC approach.	y	y	y	y	y	y	y	y	y	y
(2). The best results outperform fully fine-tuned PLMs.	y	y	y	y	y	y	y	y	y	y
(3). The best results set a new SOTA.	y	y	y	y	y	y	y	n	n	n
*(4). Using SeqC, LLMs outperform a fully fine-tuned RoBERTa.	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{8}{8}$	$\frac{8}{8}$	$\frac{5}{8}$	$\frac{2}{8}$	$\frac{3}{8}$	$\frac{4}{8}$
RQ2: Which LLMs are more effective as classifiers?										
(1). The 13B llama2 models or the 8B llama3 models produce the best results.	y	y	y	y	y	y	y	y	y	y
*(2). Using Gen, instruction-fine-tuned LLMs outperform their non-instruction-fine-tuned counterparts.	n						y			
RQ3: Which approach is most effective?										
*(1). In terms of performance, SeqC (and XNet) is most effective.										
a. SeqC outperforms Gen	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{8}{8}$	$\frac{8}{8}$	$\frac{7}{8}$	$\frac{7}{8}$
b. SeqC outperforms SNet	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{8}{8}$	$\frac{7}{8}$	$\frac{8}{8}$	$\frac{8}{8}$	-	-	-	-
c. XNet outperforms SNet	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{8}{8}$	$\frac{8}{8}$	-	-	-	-
d. In terms of performance, there are no significant differences between SeqC and XNet.	y	y	y	y	y	y	-	-	-	-
Metric	eff.	AIR	eff.	AIR	eff.	AIR	eff.	AIR	eff.	AIR
*(2). In terms of inference efficiency (eff.), SeqC is most effective.										
a. SeqC > Gen	$\frac{8}{8}$		$\frac{8}{8}$		$\frac{8}{8}$		$\frac{8}{8}$		$\frac{8}{8}$	
b. SeqC > SNet	$\frac{8}{8}$		$\frac{8}{8}$		$\frac{8}{8}$		-		-	
c. SeqC > XNet	$\frac{7}{8}$		$\frac{8}{8}$		$\frac{6}{8}$		-		-	
d. XNet > SNet	$\frac{8}{8}$		$\frac{8}{8}$		$\frac{8}{8}$		-		-	
(3). Generative models encounter AIR issues even after fine-tuning.	$\frac{0}{8}$		$\frac{1}{8}$		$\frac{1}{8}$		$\frac{6}{8}$		$\frac{3}{8}$	

Table 10: Summary of research questions, findings, and results of significance tests. A "y" indicates that the finding generalizes to the respective task, while the values represent the count of LLMs among the eight tested that support the statement. Statements marked with an asterisk (*) indicate that significance tests are applicable, and the dark green color denotes significant support for the statistical tests with p<0.05. Statements regarding model performance are evaluated based on accuracy (acc.) and macro average F1 scores (m. f1), respectively. The yellow color indicates that the performance is close to SOTA, or that the statement is supported by >= 50% of the LLMs. Further details are provided in §C.3

QLoRA (Dettmers et al., 2023), maintaining the same LoRA rank (r), LoRA alpha (a), and dropout (d) parameters as we used for the EIC task (see §B). However, with the smaller SDQ dataset, the results are consistently suboptimal; we thus adjust the parameters for the SDQ dataset as follows: for approach *Gen*, we set r=64, a=64, d=0.1; for approaches *SeqC*, *SNet*, and *XNet*, the settings are r=32, a=32, d=0.1.

On the binary pair classification datasets, the models are fine-tuned for ten epochs on the training set, with checkpoints saved after each epoch. For the larger EC and TSEC datasets, we fine-tune the models for five epochs. The final model checkpoint is selected based on evaluation results from the validation set, and the model's performance is subsequently assessed on the test set. The computational resources utilized are the same as those employed for the EIC task, as reported in §B.

C.3 Results and Discussion

Table 9 presents the results for the five additional tasks using the four approaches with the eight LLMs and two PLMs.[11] Previous SOTA results, as of Sept. 2024, are reported according to the

[11]For each approach, we utilize the best input and transformation function settings observed from the EIC task for LLMs and PLMs fine-tuning. For *Gen*, we employ ② *inst* + *structured input*. For *SeqC*, *XNet*, and *SNet*, structured inputs are used for pair classification datasets, while simple natural inputs are used for single-input datasets. The transformation function *diffABS* is applied for both *XNet* and *SNet*.

MTEB benchmark leaderboard[12] and the SOTA model evaluation results.[13] Based on these results, we address the following three research questions to determine whether the observations from the EIC task generalize to the new tasks. Table 10 summarizes the research questions, findings, and results of the significance tests.

RQ1: Are fine-tuned LLMs good classifiers? It is confirmed in the additional tasks that LLMs can be effectively enhanced to serve as good classifiers, achieving results that are better than or comparable to those of previous SOTA models and fully fine-tuned PLMs. First, we observe that across all datasets, the best results (highlighted in bold in Table 9) are achieved by fine-tuning LLMs using the *SeqC* approach. These results outperform fully fine-tuned PLMs across all datasets and establish new SOTA performance for the pair classification tasks. In the single-input tasks, our best results are comparable to the SOTA results. Next, we examine the fine-grained results for each LLM. Using the *SeqC* approach on the pair classification datasets, most LLMs outperform the fully fine-tuned RoBERTa. One-sample t-tests indicate that the higher-performing LLMs are significantly better than the RoBERTa baseline. These findings suggest that LLMs possess superior encoding capabilities and can be fine-tuned as good classifiers using the encoding-based *SeqC* approach, with particularly substantial potential for pair classification tasks.

RQ2: Which LLMs are more effective as classifiers? As shown in Table 9, across all five additional tasks, the best results (highlighted in bold) are achieved by the 13B Llama2 or 8B Llama3 models, reinforcing the findings from the EIC task. Additionally, we observe that using the *Gen* approach for the multi-class single-input classification tasks, instruction-fine-tuned LLMs consistently outperform their non-instruction-fine-tuned counterparts, with statistical significance supported by one-sided Wilcoxon signed-rank tests. However, no consistent or statistically significant performance differences are observed between the chat and non-chat versions of LLMs in binary pair classification tasks, particularly when the tasks are relatively straightforward and the label categories ('yes' or 'no') are easy to interpret. These findings suggest that instruction-fine-tuned LLMs demonstrate superiority in the *Gen* approach when dealing with complex label tags and tasks, likely due to their enhanced language comprehension capabilities.

RQ3: Which approach is most effective? The additional experiments confirm that the *SeqC* approach is superior in terms of performance, answer inclusion rate (AIR), and inference efficiency.

In terms of performance, *SeqC* and *XNet* are superior, as indicated by the four sub-statements in RQ3 (1) in Table 10, with significance supported by one-sided Wilcoxon signed-rank tests (a-c) and two-sample t-tests (d) on most datasets. Additionally, the statistical tests conducted on the overall results of all tasks, including those from EIC, provide significant evidence supporting this finding.

In terms of inference efficiency, the *SeqC* approach is significantly superior, as demonstrated in Table 11 and supported by one-sided Wilcoxon signed-rank tests across all datasets (see RQ3 (2) in Table 10). Additionally, the statistical tests on the overall results of all tasks, including those from EIC, confirm the significance of this superiority.

In four of the five datasets, we observe that LLMs continue to face AIR issues even after fine-tuning with the *Gen* approach. This issue is particularly pronounced in datasets with complex label tags. These findings suggest that the generation-based approach is not optimal in practice due to its lack of robustness, difficulty in control, and inefficiency during inference. In contrast, the proposed encoding-based approaches, particularly *SeqC*, demonstrate superiority not only in performance but also in AIR and efficiency, making them well-suited for large-scale applications.

In conclusion, the generalization evaluations conducted across the five tasks provide compelling evidence that: (1) LLMs can be fine-tuned to operate as good classifiers, achieving SOTA results; (2) among the eight tested LLMs, the 13B Llama2 and 8B Llama3 models exhibit the greatest potential; and (3) the encoding-based *SeqC* approach proves to be the most effective, demonstrating significantly superior performance, inference efficiency, and perfect AIR.

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

[12] https://huggingface.co/spaces/mteb/leaderboard

[13] https://huggingface.co/nvidia/NV-Embed-v2

Task Type	Binary Pair Class.			Single Input Class.	
Approach/Dataset	SDQ	TSE	TUC	EC	TSEC
Gen	4.5	4.2	3.3	3.5	3.3
SeqC	**19.5**	**24.6**	**20.6**	**7.7**	**8.8**
SNet	3.7	4.4	4.6	-	-
XNet	7.9	8.9	7.3	-	-

Table 11: Inference efficiency comparison across approaches. Inference efficiency is measured by the number of samples processed per second. Reported values are the average efficiency of the eight tested LLMs.

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 by the creator of the original Re3-Sci dataset, 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 12 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[14]:

- 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.
- 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.

[14] https://f1000research.com/, as of April 2024

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 12: Human evaluation of the annotated *Re3-Sci2.0* dataset. 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 13: 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.

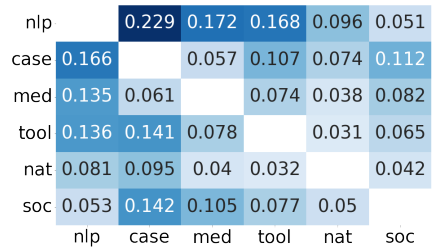
E Edit Analysis

E.1 Successful vs. Unsuccessful Revisions

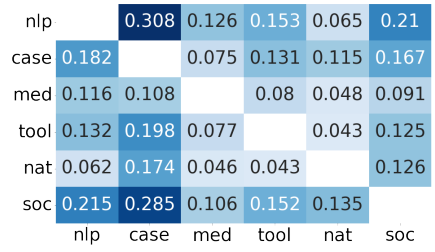
We interpret increased reviewer scores as indicators of successful revisions and improvements in scientific quality. Reviewers in the F1000RD community evaluate publications using one of three decisions: "reject," "approve-with-reservations," or "approve", which we convert into numeric values.[15] Document 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 13 shows that successful revisions contain significantly more edits than unsuccessful ones, particularly with more changes in Clarity and Claim.

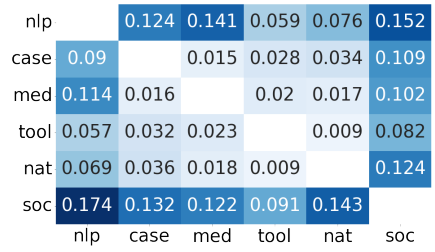
E.2 Editing Behavior across Research Domains and Document Categories



(a) action location



(b) intent location



(c) label combination

Figure 6: Kullback–Leibler (KL) Divergence analysis of the distributions across categories for (a) action location (Figure 4, 1st line) (b) intent location (Figure 4, 2nd line) and (c) edit action and intent combinations (Figure 5). The higher the KL divergence, the greater the difference between the distributions.

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