LEARN: on the feasibility of Learner Error AutoRegressive Neural annotation

Paolo Gajo[1], Daniele Polizzi[1], Adriano Ferraresi[1] and Alberto Barrón-Cedeño[1,*]

[1]Università di Bologna, Corso della Repubblica, 136, 47121, Forlì, Italy

Abstract

Error annotation is a defining feature of learner corpora, essential for understanding second-language development. Its centrality is mirrored by the meticulous effort required for its implementation, which is typically conducted in manual fashion. In this exploratory study, we investigate the feasibility of automating the task by training large language models (LLMs) in the context of dialogue-based Computer-Assisted Language Learning (CALL). We experiment with instruction-tuned LLMs across annotation granularities and prompting strategies. Results show that coarse-grained tags are more reliably predicted than fine-grained ones, with few-shot example-based prompting outperforming context-only formats. These findings point to the potential of LLMs for semi-automatic error annotation, while underscoring the need for larger datasets and the effectiveness of training models through causal LM to handle rare linguistic phenomena. Code and data: https://github.com/paolo-gajo/LEARN

Keywords

large language models, low-rank adaptation, error annotation, learner corpora, human-computer interaction

1. Introduction

Error annotation plays a crucial role in learner corpus research, a domain of inquiry that, while closely related to second language acquisition (SLA), is distinguished by its focus on providing insights into learners' interlanguage systems and acquisition patterns. The underlying assumption is that errors, defined as the application of an internalised rule not prescribed by established linguistic norms [1], are not merely indicators of textual quality, but a reflection of learners' evolving competence in their target language [2].

Regardless of the taxonomy's level of granularity, error annotation remains a time-consuming task, susceptible to inconsistencies in human judgment and inaccuracies from automatic parsers originally designed for native input [3]. As generative AI architectures begin to populate linguistic toolkits [4] and mimic established approaches to language analysis [5], an opportunity arises to reduce the burden of manual annotation while retaining the depth of linguistic insight traditionally required for this complex task. While a limited number of studies do investigate the use of the technology to annotate pragmatic and discourse-level features, including [6] on apologetic expressions and [7] on evaluative stance, its applications in the context of learner corpus research remain scarce.

To address this issue, we investigate the feasibility of training large language models (LLMs) to automate error annotation, establishing a baseline for comparison while focusing on an increasingly relevant mode of text production: human-computer interactions [8]. The task proves particularly challenging due to the complexity of the tagset adopted, the model's limited domain-specific expertise, and the scarcity of annotated training data available. Our contributions are two-fold: (*i*) We release a novel dataset containing 2,675 manual annotations of linguistic errors across fifty texts. (*ii*) Using LoRA-tuned LLMs, we assess the impact of four combinations of prompting strategies on automatic error annotation in human-computer written interactions, establishing a benchmark for future work in the area.

The rest of the paper is structured as follows: Section 2 outlines the role of learner corpora in SLA research, with a focus on error annotation practices. Section 3 introduces the dataset and the tagset used in the experiments, along with a description of the annotation process. Section 4 provides specifics on the model architecture, training, and evaluation. Section 5 lays out the settings approached for the automatic annotation task. Section 6 reports the results of the experiments. Finally, Section 7 draws conclusions and offers suggestions on future research avenues. In Appendix A, we provide a full list of the used categories and tags. Appendix B reports the full results. Appendix C provides information on the used computational resources.

CLiC-it 2025: Eleventh Italian Conference on Computational Linguistics, September 24 — 26, 2025, Cagliari, Italy

*Corresponding author.

paolo.gajo2@unibo.it (P. Gajo); daniele.polizzi2@unibo.it (D. Polizzi); adriano.ferraresi@unibo.it (A. Ferraresi); a.barron@unibo.it (A. Barrón-Cedeño)

https://www.unibo.it/sitoweb/paolo.gajo2 (P. Gajo); https://www.unibo.it/sitoweb/daniele.polizzi2 (D. Polizzi); https://www.unibo.it/sitoweb/adriano.ferraresi/cv-en (A. Ferraresi); https://www.unibo.it/sitoweb/a.barron (A. Barrón-Cedeño)

0009-0009-9372-3323 (P. Gajo); 0009-0007-1927-4158 (D. Polizzi); 0000-0002-6957-0605 (A. Ferraresi); 0000-0003-4719-3420 (A. Barrón-Cedeño)

2. Background and Motivation

Learner corpora are systematic collections of electronic texts whose key defining feature lies in the representation of "language as produced by foreign or second language (L2) learners" [9]. They are increasingly used in various strands of empirical SLA research, varying across multiple dimensions: medium (spoken or written), genre (such as essays, summaries and interviews), learners' linguistic background, sampling strategies (synchronic, longitudinal or quasi-longitudinal), intended pedagogical or research purpose, and geographical scope of data collection (ranging from local to large-scale initiatives) [9]. Each of these design parameters shapes the corpus analytical potential and determines its suitability for different lines of linguistic inquiry, particularly those aimed at identifying developmental trajectories and persistent learner difficulties [10]. Their structured format also makes them a valuable resource for the development of natural language processing (NLP) applications grounded in authentic data that are used for educational purposes [11].

Central to all of these applications is the identification and classification of errors, which serve not only as indicators of language proficiency but also as windows into the evolving interlanguage systems of learners. These errors are signalled using a predefined taxonomy that serves the purpose of assigning tags, i.e. labels capturing specific categories and subcategories of errors, to the corresponding portion of text. To ensure consistency, annotation typically follows detailed guidelines, which provide operational definitions and prototypical cases for each tag. However, the process still requires annotators to formulate a hypothesis about the nature of each error, interpreting the distance between the learner's production and the expected target form as either structural or linguistic per se [2].

In spite of the subjectivity inherently built into the task, expert judgment has so far offered the most reliable means of ensuring both consistency and linguistic accuracy, striking a delicate balance between introspection and methodological rigour that underpins high-quality learner corpus annotation. While projects like the Cambridge Learner Corpus (CLC)[1] and the International Corpus of Learner English (ICLE)[2] have demonstrated the value of error-tagged data for SLA research, annotation remains labour-intensive and demands substantial expertise and time investment. The existence of automatic approaches to learner corpus error annotation, by contrast, remains largely limited. Although some research has investigated advanced technologies such as LLMs for grammatical error identification [12], to the best of our knowledge no published work has explored their capacity to perform full-fledged annotation of learner language.

This challenge is not just one of scale, but also of scope. Learner corpora are still predominantly focused on argumentative or academic writing, mirroring the types of structured tasks performed in *traditional* educational settings. Interactive language use, by contrast, remains significantly underrepresented and tied to semi-structured interview formats [13], which only partially capture the dynamic and co-constructed nature of real-time communication. This gap is particularly problematic given the centrality of interactionist approaches to SLA, which emphasise the role of input, opportunity for output, feedback, and negotiation of meaning in driving acquisition [14]. As Granger [15] forecasts, the future of learner corpus research lies not only in enhancing annotation practices but also in expanding corpora to new educational contexts, each potentially introducing distinct patterns of learner language that call for targeted annotation strategies.

Shifts towards greater variability in learner data amplify the need for scalable, adaptive annotation methods. Our contribution presents an exploratory case study investigating whether small-scale, open-weight LLMs can reliably be trained to automate learner error annotation, evaluating not only their diagnostic capabilities but also their alignment with linguistic taxonomies and established error annotation conventions. More specifically, we test this feasibility in an unconventional setting for learner corpora annotation: informal dialogue practice.

3. Data

The dataset employed contains human–machine written interaction data, contributing to an increasingly relevant research strand focusing on conversational AI's effectiveness for language development [14]. It features English-as-foreign-language (EFL) productions of Italian university students aged 18–25 from diverse degree programs, most of whom self-report a low-to upper-intermediate proficiency level. One distinct interaction for each student (50 in total) was collected based on a protocol combining one of two different LLM-based chatbots with two EFL learning scenarios. The chatbots used during the experimental sessions are ChatGPT,[3] a general-purpose Generative AI tool, and Pi.ai,[4] a task-oriented chatbot specifically developed to engage in natural language conversation. The learning scenarios are structured around two communicative formats that constitute part of standardised English proficiency tests: open-ended conversation (small talk) and target-oriented dialogue (role play). While small talk allows participants to freely express themselves on past experiences, current interests and events or future projects, role playing requires them to

[1] https://www.cambridge.org/elt/corpus/learner_corpus2.htm
[2] https://www.uclouvain.be/en/research-institutes/ilc/cecl/icle
[3] https://chatgpt.com/
[4] https://pi.ai/talk

Source	Token Count
Learner-Produced (total)	17,730
Small talk	10,548
Role play	7,182
Chatbot-Generated (total)	95,320
Small talk	39,033
Role play	56,287
Total	**113,901**

Table 1
Dataset token distribution by task type.

use context-sensitive vocabulary and formulaic language. As such, both tasks prove particularly effective in covering a wide variety of use cases where multiple examples of errors might appear, ranging from grammar and lexis to register and style. The dataset annotation scheme features structural information on turns and contextual information on the chatbot used, the tasks performed and the learner profile. Token counts are reported in Table 1.

3.1. Tagset

Our benchmark for automatic error identification consists of fifty texts manually annotated by two expert anglicists, using an adapted version of the Louvain Error Tagging Manual Version 2.0 [16]. While the taxonomy does not align with any specific formal SLA theory or L1–L2 pairing, it was selected precisely for its broad recognition within the learner corpus research community, a *de facto* standard providing a comprehensive mapping of errors discussed in the field. The adaptation was carried out through preliminary pilot tests and includes several fine-tuning operations that introduce revised use cases and five new tags. The updated manual comprises 59 categories, spanning across eight domains: digitally-mediated communication (DMC), form (F), punctuation (Q), grammar (G), lexico-grammar (X), lexis (L), word (W), infelicities (Z) and code-switching (CS).

A subset of cases previously assigned to the category of formal errors, "unwarranted use of mother-tongue words" [16], constitutes now a separate category: namely, that of intra- or inter-sentential code-switching. The split was essential to distinguish between involuntary deviations from the expected spelling norm (covered by F, along with morphological errors in derivational affixes) and explicit cases of L1 interference as a coping mechanism in second-language communication. In a similar fashion, all instances of missing capitalisation, including lowercase letters at the beginning of a conversational turn, were assigned to DMC to capture features of texting that likely reflect the informal nature of the task rather than language competence alone. These also include abbreviations commonly found in the context of instant messaging, such as BTW or LOL. Finally, neologisms and calques have been assigned a distinct subcategory (LWCO) falling within that of lexis (L) rather than form (F). The rationale behind this change follows on Cervini and Paone's [17] classification of intercomprehension strategies, where both calques and neologisms are conceived as pertaining to the lexical dimension of communication. The remaining macro-categories are retained as originally defined [16]. Grammatical Errors (G) are violations of standard grammar rules that affect syntactic structure, including subject–verb agreement, misuse of tenses, article errors, or problems with word forms, such as pronouns and determiners. Lexico-Grammatical Errors (X) involve combination patterns specific to the word rather than sentence-wide grammar, including dependent prepositions or verb complementations. Lexical Errors (L) concern vocabulary choices that do not match the intended meaning or context, hence coming across as semantically awkward or stylistically inappropriate. Word Errors (W) target imbalances in a sentence caused by omitting necessary words, adding superfluous ones, or placing words in an unnatural or incorrect order. Punctuation Errors (Q) cover incorrect, missing, or excessive use of marks, such as commas, periods, or colons. Finally, Infelicities (Z) address stylistic concerns that, while not strictly errors, may require reformulation for the sake of clarity or naturalness (Z). See Table 8 in Appendix A for a complete list of the tags used, together with a brief description of their coverage for each use case.

Errors were marked using inline XML-style tags of the format `<TAG corr="correction">`*incorrect text*`</TAG>` via the Université Catholique de Louvain Error Tagging Editor (UCLEE).[5] In case of the addition of missing words or the omission of redundant ones, the format is `<TAG corr="correction">\0</TAG>` or `<TAG corr="\0">`*incorrect text*`</TAG>`, respectively. The software supports the insertion, editing and processing of error tags using a preferred tagset. To accommodate the specific requirements of our task, we uploaded a custom .tag file reflecting the necessary modifications we had implemented. A truncated example of file annotation can be found in Figure 1.

In line with the Louvain Manual, corrections were minimal and hypothesis-driven, ensuring that tags reflect plausible learner intentions and do not result in speculative rewriting of the original text. Tags were assigned based on the erroneous form itself, using the shortest possible span required to isolate it. Regional spelling variants (e.g., British and American English) were not flagged, as participants received no instruction on preferred norms. Likewise, punctuation errors were annotated only when they hindered readability, in recognition of informal communication habits. Cases where multiple errors overlapped were nested within one another, with

[5] https://oer.uclouvain.be/jspui/handle/20.500.12279/968

```
<?xml version="1.0" encoding="utf-8"?>
<file name="id_1.txt" tagset="uclee-en-2.0.tag"> <text
id="id_1" area_of_study="Social sciences" age="24" [...]>
    <task type="small talk">
        <turn type="chatbot" who="Pi.ai">Hey there, great to
        meet you. I'm Pi, your personal AI. [...]</turn>
        <turn type="student">Hi</turn>
        <turn type="chatbot" who="Pi.ai">Hey User!
        How's everything going on your side? [...]</turn>
        <turn type="student"><DMCC
        corr="How">how</DMCC> are you today?</turn>
        [...]
    </task>
    <task type="role play">
        <turn type="student"><DMCC
        corr="You">you</DMCC> are an encouraging tutor
        who helps students improve their <DMCC
        corr="English">english</DMCC> by engaging in role
        play <FS corr="activities">actvities</FS>.>[...]</turn>
        <turn type="chatbot" who="Pi.ai">Great idea! Let's
        start the role play. As the Restorative Justice, I'm
        interested in [...]</turn>
        [...]
    </task>
</text>
</file>
```

Figure 1: XML annotation output of the UCLEE software.

Table 2
Distribution of the tags in the data used for training, development, and testing.

Tag	#						
DMCC	927	LP	45	GDO	13	XNCO	4
FS	314	LSV	45	XNUC	12	XADJCO	4
GA	149	LSN	43	QR	12	GPD	3
LSPR	80	CSINTRA	33	CSINTER	11	LCC	3
GNN	80	GVN	32	GPI	10	LCLC	3
GPP	72	XVPR	28	GADVO	9	GADJO	2
GVT	64	GNC	27	GDI	8	XPRCO	2
WO	63	QC	24	GDT	8	GPU	2
QM	60	GVNF	23	GADJCS	7	GPO	2
Z	54	DMCA	23	XNPR	7	XADVPR	1
LWCO	52	GPR	20	GDD	6	LCLS	1
XVCO	51	GVM	18	QL	6	GPF	1
WM	51	LSADV	16	FM	5		
GVAUX	51	GWC	15	XADJPR	5		
WR	49	LSADJ	15	LCS	4		

spelling errors being considered the lowest level, i.e. the first correction to be applied.

Inter-annotator agreement (IAA) was calculated on five separate texts using the Gamma coefficient [18], a metric suited to evaluating categorical labels with overlapping text spans. Annotation files were first parsed to extract error tags and their corresponding character offsets using a custom XML processing function. The agreement was recorded only when annotators applied the same error tag to mark the exact same character span as erroneous. Scores registered a mean of 0.77024 ± 0.09270. The computation was repeated a second time on all tags except those targeting formal spelling (FS) and digitally-mediated communication (DMC). That is, taking into account the most subjective among the sub-categories in our tagset, which account for 53.60% of all the tagged issues. The results show an agreement of 0.74698 ± 0.13027. Given the strictness of our criteria, we consider the obtained IAA to be highly satisfactory and reliable, since $\gamma < 0$ signifies worse-than-random agreement and the upper bound is $\gamma = 1$.

3.2. Data processing

The data are compiled by filtering out the chatbot responses and splitting the collection into training, development, and testing partitions with an 80/10/10 split. Five different (fixed) seeds are used to split the data and initialise model states, which helps us mitigate variance in the results. Table 2 provides information on the distribution of the tags, which has a long tail formed by rare tags, 22 of which have fewer than 10 occurrences.

As exemplified in Figure 2, we experiment with two types of in-context learning (ICL) sections (bottom row), each using fine- or coarse-grained tags (top row), for a total of four prompt combinations. The prompt starts with a system message defining the LLM persona, followed by the instruction. The macro categories or tags are then optionally listed. In the first experimental setting, a varying number of ICL examples is included. For all data splits, pairs of examples are sampled at random solely from the training set, across any of the student-chatbot conversations. We sample an equal number of examples with and without error annotations.[6] Finally, the task is repeated to mark the target sentence.

In the second setting, we provide the model with the context of the conversation to which the target message belongs. Note that in this case, what we divide in 80/20/20 splits is the list of conversations, rather than the individual messages. Since conversations do not all have the same size, in this case each seed produces different split sizes, as shown in Table 3.

In our experiments, we wish to showcase the impact of using random annotated instances vs unannotated context. Therefore, although the data partitions used in the two settings are produced in different ways, we still deem our approach to be valid, considering the use of five different seeds.

[6]The original and the annotated utterances are separated by ### symbols to avoid any subwords being merged with the separator by the used tokenisers.

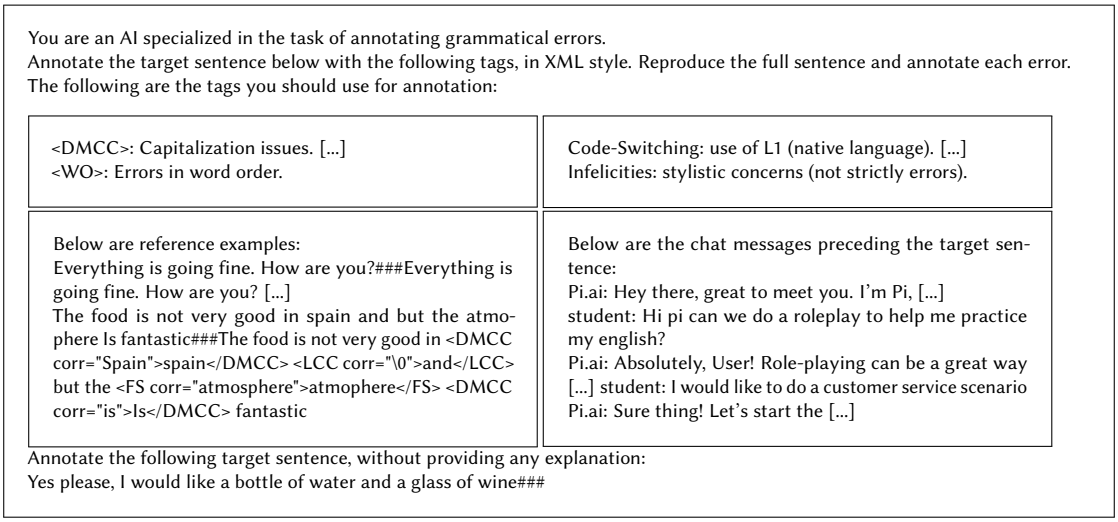

Figure 2: Prompt example with fine-grained tags (top left) or coarse categories (top right), followed by either randomly sampled pairs of examples (bottom left) or previous chat context (bottom right). All four combinations are possible.

Table 3
Split sizes for the training, development, and testing partitions, for the random ICL sampling and context prompt settings.

Setting	Train	Dev	Test
Rng ICL	831	104	104
Context	822.6 ± 11.586	109.0 ± 14.656	107.4 ± 11.740

4. Model

For our experiments, we adopt pre-trained decoder-only Transformer [19] models of the LLaMA 3 series [20], publicly available through Hugging Face.[7] The models we choose are first pre-trained on large unstructured corpora and then fine-tuned on instruction prompts with a causal language modeling objective (NLL):

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log(p_\theta(w_j|w_{<j})). \tag{1}$$

Then, they are instruction-tuned through supervised fine-tuning and reinforcement learning from human feedback using direct policy optimization [20]. This effectively makes them chatbots capable of fulfilling user requests.

We fine-tune the models with the same objective as in Eq. (1) on the prompts as described in Section 3.2.[8] We calculate the loss for both the prompt and the completion, since we want the model to learn to predict the annotated sentences not just from the target sentence, but also from the tags and the examples included in the prompt. In other words, we simultaneously train the model on a large amount of sampled examples within the prompt, through teacher forcing, and we also instruction-tune it to predict the desired target sentence.

The architecture of these models consists in a token/positional embedding layer, followed by a stack of decoders, with a language modeling classifier on top. Each decoder comprises a grouped-query attention layer [21], followed by a set of MLP layers each using a SwiGLU activation function [22]. We update the weights of the decoder blocks with LoRA [23], only targeting the key, query, and value matrices Q, K, V of the attention layers:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right)V$$

where M is the matrix filled with zero values in the lower triangular part and $-\infty$ elsewhere, and d_k is the output dimension of Q and K. The attention and MLP layer parameters are kept frozen during training. The original input to these layers is simultaneously processed through LoRA components consisting of weight matrices $B \in \mathbb{R}^{d_i \times r}$ and $A \in \mathbb{R}^{r \times d_o}$, where $r \ll d_1, d_2$. Here, r represents the low-rank projection dimension, while d_1 and d_2 correspond to the input and output dimensions of each respective layer. During training, only the LoRA matrices B and A receive parameter updates. Thus, the forward pass of an input $\mathbf{x}$ through an MLP with frozen weight W_0 is modified as:

[7] https://huggingface.co

[8] The model needs to be given the prompt in a chat template (https://huggingface.co/docs/transformers/en/chat_templating#applychattemplate) which we omit here for clarity.

$$W_0\mathbf{x} + \frac{\alpha}{r}BA\mathbf{x} = (W_0 + \Delta W)\mathbf{x} = W_1\mathbf{x}$$

The scalar α acts similarly to the learning rate adjustment provided by the Adam optimizer [24], according to [23]. Each module combines the outputs of the frozen layer and its corresponding LoRA layer through element-wise addition. We initialize the LoRA blocks using $r = \alpha = 16$, without biases or dropout.

We train for 3 epochs using a batch size of 4, without gradient accumulation. We employ a learning rate of 2×10^{-4} with 5 warm-up steps, weight decay of 0.01, and AdamW [25] as the optimization algorithm. Prior to fine-tuning, *Llama-3.3-70B-Instruct* is quantized at 4-bit precision with QLoRA [26], using bitsandbytes.[9]

Due to the sparsity of low-occurrence tags, we focus on evaluating the model on the most common ones using micro-averaged precision, recall, and F_1-measure. The prediction of a tag is considered correct only if both the tag and the associated text match. For example, in the sentence `<DMCC corr="Not">not</DMCC> really, what is your proposal <QM corr="?">\0</QM>` the prediction would be incorrect if the tag DMCC was assigned to "not really" rather than just "not". As regards this example, also note that the model is required to generate "\0" tokens, representing omitted words.

Each model is fine-tuned and evaluated on five different seeds, for which we report the average performance along with the standard deviation. During evaluation, we allow the model to generate up to 1,000 new tokens, which we deem sufficient based on instance lengths. We select the best epoch based on the highest micro-averaged F_1-measure on the development set. We report micro-averaged metrics, since macro-averaging does not provide a faithful picture of model performance, due to the long tail of low-occurrence classes (Table 2).

5. Experiments

We task the fine-tuned models to automatically annotate linguistic errors in sentences written by learners of English. We experiment with two levels of granularity of error classification, one at the level of the macro category (e.g., "Form", or "Punctuation") and one at the tag level, i.e. those listed in Table 2.

We also use two different types of prompts. The first includes $N_{\text{ICL}} \in \{0, 2, 4, 6, 8, 10\}$ pairs of unannotated and annotated student messages. We vary the number because an insufficient amount might not provide the model with enough information to produce optimal performance, while an excessive quantity might excessively shift attention from the target task. The second type of prompt includes the $k = 10$ chat messages preceding the student message that the model is tasked to annotate.

We use *Llama-3.1-8B-Instruct* to first conduct a hyperparameter search as regards the number of in-context learning examples to use and whether to include the tags in the prompt. Then, we use the bigger *Llama-3.3-70B-Instruct* with the best combination of hyperparameters.

[9] https://github.com/bitsandbytes-foundation/bitsandbytes

Table 4
Overall micro-averaged results for *Llama-3.1-8B-Instruct* and *Llama-3.3-70B-Instruct* on the fine-grained classification task, using randomly sampled ICL examples. Best in bold.

Tags	N_{ICL}	F_1	Precision	Recall
Llama-3.1-8B-Instruct				
×	0	0.397 ±0.034	0.435 ±0.051	0.367 ±0.025
	2	0.416 ±0.040	0.427 ±0.053	0.407 ±0.038
	4	**0.424** ±0.029	0.431 ±0.036	0.419 ±0.026
	6	**0.424** ±0.023	0.421 ±0.030	**0.427** ±0.018
	8	0.412 ±0.022	0.407 ±0.028	0.417 ±0.016
	10	0.405 ±0.045	0.403 ±0.048	0.407 ±0.044
✓	0	0.377 ±0.043	0.425 ±0.063	0.341 ±0.035
	2	0.421 ±0.041	**0.440** ±0.048	0.405 ±0.041
	4	0.401 ±0.035	0.420 ±0.041	0.384 ±0.036
	6	0.399 ±0.025	0.412 ±0.043	0.388 ±0.016
	8	0.407 ±0.050	0.400 ±0.061	0.415 ±0.040
	10	0.399 ±0.028	0.401 ±0.039	0.399 ±0.019
Llama-3.3-70B-Instruct				
×	6	0.472 ±0.029	0.470 ±0.027	0.476 ±0.034

Table 5
Overall micro-averaged results for *Llama-3.1-8B-Instruct* and *Llama-3.3-70B-Instruct* on the coarse-grained classification task, using randomly sampled ICL examples. Best in bold.

Tags	N_{ICL}	F_1	Precision	Recall
Llama-3.1-8B-Instruct				
×	0	0.440 ±0.024	0.397 ±0.014	**0.494** ±0.044
	2	0.439 ±0.033	0.434 ±0.036	0.445 ±0.037
	4	0.450 ±0.030	0.436 ±0.033	0.467 ±0.041
	6	0.460 ±0.047	0.451 ±0.050	0.470 ±0.047
	8	0.436 ±0.037	0.437 ±0.029	0.435 ±0.046
	10	0.446 ±0.035	0.446 ±0.036	0.448 ±0.050
✓	0	0.424 ±0.031	0.382 ±0.031	0.478 ±0.042
	2	0.456 ±0.044	0.437 ±0.043	0.477 ±0.045
	4	0.440 ±0.030	0.454 ±0.035	0.432 ±0.056
	6	**0.466** ±0.018	**0.464** ±0.026	0.469 ±0.014
	8	0.449 ±0.033	0.463 ±0.032	0.436 ±0.036
	10	0.449 ±0.050	0.451 ±0.047	0.448 ±0.058
Llama-3.3-70B-Instruct				
✓	6	0.502 ±0.024	0.514 ±0.037	0.492 ±0.031

6. Results

Random sampling ICL The results marginalised across all classes for the fine-grained setting are listed in Table 4. The best performance is achieved with $N_{\text{ICL}} = 6$

pairs of examples, 6 positive and 6 negative. This shows our concerns with finding the best number of examples were founded, since higher amounts lead to increasingly worse performance. However, most of the performance gain is obtained by going from $N_{\text{ICL}} = 0$ to even just providing 2 pairs of examples, even without the model being shown the meaning of the tags. Indeed, overall the best results for *Llama-3.1-8B-Instruct* are achieved when not including the tags and their descriptions in the prompt. Gajo and Barrón-Cedeño [27] report similar results, where increasing the number of examples yielded diminishing returns when extracting RDF triples from texts and overly long lists of references in the prompt diluted model attention away from the target task.

Fine-tuning *Llama-3.3-70B-Instruct* with the best hyperparameter $N_{\text{ICL}} = 6$ and no tags in the prompt, the model obtains a micro-F_1 of 0.472. Out of five seeds, the highest validation performance is obtained twice on the first epoch, twice on the second, and only once on the third. Since the model is only shown 831 training examples and the first and second epochs already provide the best performance, the model seems to fit very quickly to the patterns it needs to recognize to identify errors.

The overall results for the coarse-grained categories are reported in Table 5. The performance is overall slightly higher when including the categories in the prompt. In this case, since only 9 classes are listed, the model is able to make good use of the provided information. Indeed, not only are the mean scores higher, but the standard deviation is also lower at $N_{\text{ICL}} = 6$, which is the setting that yields the highest performance with *Llama-3.1-8B-Instruct*. As for *Llama-3.3-70B-Instruct*, performance is greater, but with a smaller gap between the two models, compared to the fine-grained tags.

The full results for each fine-grained tag at all values of N_{ICL} are reported in Table 9 in Appendix B. At the fine-grained level, only a few high-frequency tags such as DMCC (927 instances) and FS (314) are predicted reliably. Most of the others are either predicted with very high standard deviations or do not receive predictions at all, due to the sparsity of labels. Nonetheless, the performance for several morphosyntactic tags, e.g. GNN (80), GPP (72) and GVAUX (51) exhibits gradual improvements with increasing values of N_{ICL}, indicating that training the model on a higher number of examples might be beneficial for some classes.

Based on the distribution shown in Table 2, the amount of training instances per class indeed seems to strongly correlate with performance. However, Z (54), used to indicate stylistic problems, is never predicted correctly by either of the models, despite having a number of instances comparable to that of much better-performing classes, e.g. QM (60) or WM (51), respectively used for missing punctuation and words. Since the latter clearly affect the format and structure of the sentence via omission, this hints at the fact that the model more easily handles structural errors, compared to those where style and semantics are involved.

Table 6
Overall micro-averaged results for *Llama-3.1-8B-Instruct* and *Llama-3.3-70B-Instruct* for the context prompt setting, using fine-grained ($\mathcal{F}$) and coarse ($\mathcal{C}$) categories.

	Tags	F_1	Precision	Recall
Llama-3.1-8B-Instruct				
$\mathcal{F}$	×	0.221 ± 0.079	0.256 ± 0.071	0.198 ± 0.083
	✓	0.207 ± 0.091	0.237 ± 0.100	0.194 ± 0.093
$\mathcal{C}$	×	0.234 ± 0.090	0.275 ± 0.097	0.208 ± 0.088
	✓	0.186 ± 0.056	0.214 ± 0.100	0.191 ± 0.075
Llama-3.3-70B-Instruct				
$\mathcal{F}$	×	0.395 ± 0.109	0.360 ± 0.088	0.375 $pm 0.095$
$\mathcal{C}$	×	0.455 ± 0.084	0.417 ± 0.076	0.434 $pm 0.077$

Table 10 in Appendix B reports the results for each coarse-grained category for all values of N_{ICL}.

Context ICL As shown in Table 6, the performance using context prompts is much lower than when using randomly sampled example pairs. An analysis of *Llama-3.1-8B-Instruct*'s predictions shows that, at times, the model makes mistakes even on easy instances of the DMC category, i.e. the one with overall highest results. For example, in "`student: It's perfect! Thank <XVCO corr="you">u</XVCO> so much`", the model assigns XVCO (errors with verb complementation) rather than DMCA to a clear-cut case of Internet-style abbreviation. Considering the performance on this class is above 0.800 when using random ICL example pairs, this is a clear hint that the context does not provide useful information for the best-performing categories. Indeed, the macro-categories for which contextual information is likely to be most relevant are lexis (L) and infelicities (Z), where discourse-level or pragmatic cues are critical in assessing appropriateness and distinguishing genuine errors from stylistic deviations. However, as shown in Table 7, the performance for these categories is very low (L) or null (Z). For *Llama-3.1-8B-Instruct*, the performance on the L category ($F_1 = 0.070$) is worse than the one obtained in the random ICL sampling setting, even with $N_{\text{ICL}} = 0$ ($F_1 = 0.091$, see Table 10). Therefore, even in the cases in which the model would supposedly benefit from being provided the context of the conversation, simply having it memorize decontextualized examples through causal language modeling provides better performance. Indeed, as already mentioned in the previous section, the model likely pays more attention to the shallow structure of the sentence rather complex semantic relationships. Thus, having it learn annotations directly from XML-formatted examples provides superior performance. This is also

Table 7
Micro-averaged F_1 results per category for *Llama-3.1-8B-Instruct* and *Llama-3.3-70B-Instruct* with the best-performing $N_{\text{ICL}} = 6$ using coarse-grained categories. C=CS, D=DMC.

		Rng ($N_{\text{ICL}} = 6$)		**Context ($k = 10$)**	
Tags		***8B***	***70B***	***8B***	***70B***
C	×	0.050 $\pm$0.112	0.000 $\pm$0.000		
	✓	0.197 $\pm$0.192	0.175 $\pm$0.186	0.000 $\pm$0.000	0.053 $\pm$0.119
D	×	0.813 $\pm$0.059	0.512 $\pm$0.149		
	✓	0.827 $\pm$0.051	0.854 $\pm$0.036	0.552 $\pm$0.130	0.759 $\pm$0.088
F	×	0.534 $\pm$0.047	0.269 $\pm$0.088		
	✓	0.497 $\pm$0.123	0.551 $\pm$0.090	0.155 $\pm$0.060	0.433 $\pm$0.103
G	×	0.247 $\pm$0.039	0.094 $\pm$0.045		
	✓	0.306 $\pm$0.025	0.333 $\pm$0.041	0.075 $\pm$0.037	0.242 $\pm$0.061
Z	×	0.000 $\pm$0.000	0.000 $\pm$0.000		
	✓	0.000 $\pm$0.000	0.000 $\pm$0.000	0.000 $\pm$0.000	0.000 $\pm$0.000
X	×	0.068 $\pm$0.064	0.000 $\pm$0.000		
	✓	0.064 $\pm$0.095	0.117 $\pm$0.149	0.000 $\pm$0.000	0.038 $\pm$0.054
L	×	0.157 $\pm$0.054	0.065 $\pm$0.042		
	✓	0.168 $\pm$0.076	0.201 $\pm$0.051	0.070 $\pm$0.050	0.103 $\pm$0.048
Q	×	0.194 $\pm$0.129	0.000 $\pm$0.000		
	✓	0.222 $\pm$0.102	0.262 $\pm$0.102	0.000 $\pm$0.000	0.184 $\pm$0.200
W	×	0.081 $\pm$0.063	0.000 $\pm$0.000		
	✓	0.066 $\pm$0.067	0.117 $\pm$0.129	0.000 $\pm$0.000	0.050 $\pm$0.090

clear based on the fact that *Llama-3.1-8B-Instruct* can outperform its bigger counterpart just by changing the prompting strategy, although the performance obtained by *Llama-3.3-70B-Instruct* when using context prompts is closer to the one obtained with random sampling ICL.

The context ICL results for all fine-grained tags can be found in Table 11 in Appendix B.

7. Conclusions

In this study, we have built a corpus of human-computer interactions, assessing the feasibility of fine-tuning LLMs to automatically carry out error annotation. Through a series of experiments across two annotation granularities (coarse and fine-grained), we evaluated the capabilities and limitations of both *Llama-3.1-8B-Instruct* and *Llama-3.3-70B-Instruct* to learn through causal LM from two prompting paradigms. The first included the conversation context of the message requiring annotation, while the other entailed a varying number of randomly sampled ICL examples. Both prompt types optionally included explicit information about the target error classes.

Perhaps unsurprisingly, coarse-grained annotation obtains better scores than fine-grained tagging across all configurations, suggesting the viability of a hybrid, semi-automatic pipeline where LLMs handle broader error categories before finer distinctions are resolved through human post-editing or specialised tools. Model performance improved via ICL examples, peaking around 6 pairs of positive and negative instances, before exhibiting diminishing returns. This trend held across both granularities and prompt types, although not always linearly. In particular, random example-based prompts yielded substantially higher and more stable results compared to context-only ones, for both the fine- and coarse-grained annotation tasks, suggesting that focused demonstration of error-tag mappings better supports autoregressive modeling than situational grounding. The lower effectiveness of context-only prompts may also reflect a mismatch between the data and the annotation scheme, where error identification, at least of the issues observed in these conversations, is mostly self-contained within each learner's turn. Including additional text to be processed likely dilutes the model's attention, which is spread across a higher number of tokens, ultimately lowering learning effectiveness.

At a tag-specific level, results highlight the challenges of sparse class supervision for this task, with only a handful of high-frequency labels being predicted reliably. Nonetheless, we provide evidence of LLMs being able to internalise recurring learner patterns through causal LM, given they are shown enough instances.

Variation across the explored hyperparameters was modest. This implies that the performance ceilings are primarily determined by task complexity and data sparsity, rather than the suboptimal nature of specific training approaches.

In future work, we plan to produce synthetic training data for the task approached in this work, in order to improve model performance. In addition, we wish to extend the annotation to additional resources and leverage them for the development of better automatic error annotation systems. Finally, we aim to evaluate model performance also in terms of the proposed corrections.

Acknowledgments

We express our sincere gratitude to Arianna Paradisi (University of Bologna) for her valuable support and insightful contributions to the development of the error tagging manual. Her expertise and collaboration were instrumental in shaping the guidelines used in this work. We also thank the research team of the UNITE - UNiversally Inclusive Technologies to practice English[10] project for providing the resources that made this study possible.

[10]UNITE – UNiversally Inclusive Technologies to practice English (funded by the European Union – NextGenerationEU under Italy's National Recovery and Resilience Plan (PNRR); Project code 2022JB5KAL, CUP J53D23008070006

References

[1] G. Berruto, Le regole in linguistica, in: N. Grandi (Ed.), La grammatica e l'errore, Bologna University Press, Bologna, 2015, pp. 43–61.

[2] A. Lüdeling, H. Hirschmann, Error annotation systems, in: S. Granger, G. Gilquin, F. Meunier (Eds.), The Cambridge Handbook of Learner Corpus Research, Cambridge University Press, 2015, pp. 135–157. doi:10.1017/CBO9781139649414.007.

[3] G. Gilquin, Learner corpora, in: M. Paquot, S. T. Gries (Eds.), A Practical Handbook of Corpus Linguistics, Springer, Cham, 2020, pp. 283–303.

[4] L. Anthony, Corpus ai: Integrating large language models (llms) into a corpus analysis toolkit, 2023. URL: https://osf.io/srtyd/.

[5] N. Curry, P. Baker, G. Brookes, Generative ai for corpus approaches to discourse studies: A critical evaluation of chatgpt, Applied Corpus Linguistics 4 (2024) 100082.

[6] D. Yu, L. Li, H. Su, M. Fuoli, Assessing the potential of llm-assisted annotation for corpus-based pragmatics and discourse analysis: The case of apology, International Journal of Corpus Linguistics 29 (2024) 534–561. doi:10.1075/ijcl.23087.yu.

[7] M. Imamovic, S. Deilen, D. Glynn, E. Lapshinova-Koltunski, Using chatgpt for annotation of attitude within the appraisal theory: Lessons learned, in: S. Henning, M. Stede (Eds.), Proceedings of the 18th Linguistic Annotation Workshop (LAW-XVIII), Association for Computational Linguistics, St. Julians, Malta, 2024, pp. 112–123. URL: https://aclanthology.org/2024.law-1.11/.

[8] L. Kohnke, B. L. Moorhouse, D. Zou, Chatgpt for language teaching and learning, RELC Journal 54 (2023). doi:10.1177/00336882231204379.

[9] G. Gilquin, From design to collection of learner corpora, in: F. Meunier, G. Gilquin, S. Granger (Eds.), The Cambridge Handbook of Learner Corpus Research, Cambridge University Press, 2015, pp. 9–34. doi:10.1017/CBO9781139649414.002.

[10] N. Nesselhauf, Learner corpora and their potential for language teaching, in: J. M. Sinclair (Ed.), How to Use Corpora in Language Teaching, John Benjamins, 2004, pp. 125–152. doi:10.1075/scl.12.11nes.

[11] F. Meunier, Introduction to learner corpus research, in: N. Tracy-Ventura, M. Paquot (Eds.), The Routledge Handbook of Second Language Acquisition and Corpora, Routledge, New York, 2020, pp. 23–36.

[12] C. Davis, et al., Prompting open-source and commercial language models for grammatical error correction of english learner text, arXiv (2024). URL: https://doi.org/10.48550/ARXIV.2401.07702. arXiv:2401.07702.

[13] Centre for English Corpus Linguistics, Learner corpora around the world, 2024.

[14] S. Bibauw, W. Van den Noortgate, T. François, P. Desmet, Dialogue systems for language learning: A meta-analysis, Language Learning & Technology 26 (2022) 1–24. URL: https://www.lltjournal.org/item/10125-73488/.

[15] S. Granger, Learner corpora and error annotation: Where are we and where are we going?, International Journal of Learner Corpus Research 10 (2024) 25–45. doi:10.1075/ijlcr.00008.gra.

[16] S. Granger, H. Swallow, J. Thewissen, The louvain error tagging manual version 2.0, 2022. URL: https://oer.uclouvain.be/jspui/bitstream/20.500.12279/968/4/Granger%20et%20al._Error%20tagging%20manual%202.0_final_CC.pdf.

[17] C. Cervini, E. Paone, Comunicare all'universitÀ: Quando l'interazione orale si fa plurilingue, Italiano LinguaDue 16 (2024) 496–523.

[18] Y. Mathet, A. Widlöcher, J. Métivier, The unified and holistic method gamma (γ) for inter-annotator agreement measure and alignment, Computational Linguistics 41 (2015) 437–479. URL: https://doi.org/10.1162/COLI_a_00230. doi:10.1162/COLI_a_00230.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is All you Need, in: Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[20] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, The Llama 3 Herd of Models, 2024. URL: http://arxiv.org/abs/2407.21783.

[21] J. Ainslie, J. Lee-Thorp, M. d. Jong, Y. Zemlyanskiy, F. Lebrón, S. Sanghai, GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints, 2023. URL: http://arxiv.org/abs/2305.13245. doi:10.48550/arXiv.2305.13245.

[22] N. Shazeer, GLU Variants Improve Transformer, 2020. URL: http://arxiv.org/abs/2002.05202. doi:10.48550/arXiv.2002.05202.

[23] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-Rank Adaptation of Large Language Models, 2021. URL: http://arxiv.org/abs/2106.09685.

[24] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 2017. URL: http://arxiv.org/abs/1412.6980.

[25] I. Loshchilov, F. Hutter, Decoupled Weight Decay Regularization, 2019. URL: http://arxiv.org/abs/1711.05101.

[26] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettle-

moyer, Qlora: Efficient finetuning of quantized llms, arXiv preprint arXiv:2305.14314 (2023).

[27] Gajo, Barrón-Cedeño, Natural vs Programming Language in LLM Knowledge Graph Construction, Information Processing & Management 62 (2025) 104195. URL: https://www.sciencedirect.com/science/article/pii/S0306457325001360. doi:https://doi.org/10.1016/j.ipm.2025.104195.

Table 8
Categories (in italics), descriptions, and references for the error tags used in corpus annotation.

Tag	Description	Tag	Description
Digitally-Mediated Communication			
<DMCC>	Capitalization issues.	**<DMCA>**	Use of abbreviations in digitally mediated communication (e.g., OK, lol, etc.).
Form			
<FS>	Spelling errors.	**<FM>**	Morphological errors involving derivational affixes.
Punctuation			
<QM>	Missing punctuation.	**<QR>**	Redundant punctuation.
<QC>	Confusion of punctuation marks.	**<QL>**	Punctuation mark instead of lexical item (or vice versa).
Grammar			
<GDD>	Errors with demonstrative determiners.	**<GDO>**	Errors with possessive determiners.
<GDI>	Errors with indefinite determiners.	**<GDT>**	Errors with other types of determiners.
<GA>	Errors with articles (definite/indefinite/zero).	**<GADJCS>**	Errors with comparative or superlative adjectives.
<GADJN>	Errors with adjective number.	**<GADJO>**	Errors with adjective order.
<GADVO>	Misplaced adverbs.	**<GNC>**	Errors with noun case (e.g., Saxon genitive misuse).
<GNN>	Errors with noun number.	**<GPD>**	Errors with demonstrative pronouns.
<GPP>	Errors with personal pronouns.	**<GPO>**	Errors with possessive pronouns.
<GPI>	Errors with indefinite pronouns.	**<GPF>**	Errors with reflexive or reciprocal pronouns.
<GPR>	Errors with relative or interrogative pronouns.	**<GPU>**	Unclear pronominal reference.
<GVAUX>	Misuse of primary, modal, or semi-auxiliaries.	**<GVM>**	Errors with verb morphology.
<GVN>	Errors with subject-verb agreement.	**<GVNF>**	Errors in -ing, infinitives, or relative clauses.
<GVT>	Misuse of tense or aspect.	**<GVV>**	Errors with active/passive voice.
<GWC>	Confusion between word classes.		
Lexico-Grammar			
<XADJCO>	Errors with adjective complementation.	**<XNCO>**	Errors with noun complementation.
<XPRCO>	Errors with preposition complementation.	**<XVCO>**	Errors with verb complementation.
<XADJPR>	Errors with adjective-dependent prepositions.	**<XADVPR>**	Errors with adverb-dependent prepositions.
<XNPR>	Errors with noun-dependent prepositions.	**<XVPR>**	Errors with verb-dependent prepositions.
<XNUC>	Errors in uncountable/countable noun use.		
Lexis			
<LCC>	Errors in coordinating conjunctions.	**<LCS>**	Errors in subordinating conjunctions.
<LCLS>	Errors with single logical connectors.	**<LCLC>**	Errors with complex logical connectors.
<LSADJ>	Conceptual/collocational errors with adjectives.	**<LSADV>**	Conceptual/collocational errors with adverbs.
<LSN>	Conceptual/collocational errors with nouns.	**<LSPR>**	Conceptual/collocational errors with prepositions.
<LSV>	Conceptual/collocational errors with verbs.	**<LWCO>**	Coined words or calques.
<LP>	Errors in fixed word combinations, including idioms, compounds, and phrasal verbs.		
Word			
<WM>	Missing words.	**<WR>**	Redundant words.
<WO>	Word order errors.		
Code-Switching			
<CSINTRA>	Code-switching within a sentence.	**<CSINTER>**	Code-switching between sentences or turns.
Infelicities			
<Z>	Stylistic problems or unclear sequences requiring reformulation.		

A. Full list of tags

In this section, we report on the tagset used for the learner error annotation task, a revised version of the *UCLouvain Error Editor Version 2*. Table 8 lists all of the error macro- and micro-categories, their specific tags, and a brief description of each tag.

B. Full results

Here, we report the full results for *Llama-3.1-8B-Instruct* and *Llama-3.3-70B-Instruct*. The results for the random ICL sampling setting are reported in Table 9 for the fine-grained tags and in Table 10 for the coarse-grained categories. The results for the fine-grained categories in the context prompt setting are reported in Table 11.

C. Computational resources

For each prompt type, training *Llama-3.1-8B-Instruct* took ~20 minutes on a single NVIDIA H100 (96GB of VRAM), for a total of about 17 hours over all the 50 combinations of seeds and hyperparameters. Training *Llama-3.3-70B-Instruct* for each of its five runs per setting took around 90 minutes, for an additional 15 hours for the two prompt types.

Table 9
Micro-averaged F_1 results per tag for *Llama-3.1-8B-Instruct* and *Llama-3.3-70B-Instruct* in the fine-grained setting, using varying amounts of randomly-sampled pairs of ICL examples. Missing rows indicate that the model did not make any predictions.

		Llama-3.1-8B-Instruct						*70B*
	Tags	**0**	**2**	**4**	**6**	**8**	**10**	**6**
CSINTER	×	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.067 ±0.149	0.333 ±0.471	0.200 ±0.447	0.267 ±0.365
	✓	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.100 ±0.224	0.000 ±0.000	
CSINTRA	×	0.000 ±0.000	0.000 ±0.000	0.067 ±0.149	0.164 ±0.157	0.044 ±0.099	0.050 ±0.112	0.174 ±0.173
	✓	0.000 ±0.000	0.067 ±0.149	0.000 ±0.000	0.057 ±0.128	0.000 ±0.000	0.089 ±0.199	
DMCA	×	0.000 ±0.000	0.000 ±0.000	0.080 ±0.179	0.133 ±0.298	0.067 ±0.149	0.280 ±0.259	0.271 ±0.269
	✓	0.000 ±0.000	0.160 ±0.358	0.180 ±0.249	0.000 ±0.000	0.333 ±0.333	0.067 ±0.149	
DMCC	×	0.809 ±0.033	0.800 ±0.047	0.812 ±0.077	0.811 ±0.024	0.817 ±0.027	0.795 ±0.076	0.838 ±0.039
	✓	0.788 ±0.059	0.827 ±0.056	0.815 ±0.039	0.811 ±0.055	0.812 ±0.062	0.803 ±0.047	
FS	×	0.412 ±0.145	0.511 ±0.083	0.482 ±0.083	0.511 ±0.118	0.500 ±0.082	0.455 ±0.123	
	✓	0.396 ±0.065	0.453 ±0.120	0.438 ±0.077	0.442 ±0.089	0.503 ±0.045	0.477 ±0.093	
GA	×	0.068 ±0.097	0.269 ±0.053	0.252 ±0.123	0.281 ±0.098	0.223 ±0.135	0.306 ±0.124	
	✓	0.104 ±0.076	0.225 ±0.191	0.196 ±0.145	0.189 ±0.136	0.315 ±0.218	0.224 ±0.088	
GADVO	×							0.080 ±0.179
GDI	×							0.200 ±0.447
GNC	×							0.100 ±0.224
	✓	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.080 ±0.179	0.050 ±0.112	
GNN	×	0.101 ±0.096	0.101 ±0.095	0.156 ±0.104	0.191 ±0.093	0.187 ±0.171	0.176 ±0.050	0.193 ±0.131
	✓	0.117 ±0.078	0.088 ±0.050	0.092 ±0.095	0.102 ±0.060	0.144 ±0.047	0.124 ±0.130	
GPI	×	0.000 ±0.000	0.080 ±0.179	0.100 ±0.224	0.000 ±0.000	0.100 ±0.224	0.133 ±0.298	0.133 ±0.298
	✓	0.080 ±0.179	0.133 ±0.298	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.100 ±0.224	
GPP	×	0.059 ±0.054	0.230 ±0.119	0.102 ±0.144	0.264 ±0.190	0.165 ±0.027	0.199 ±0.072	0.359 ±0.207
	✓	0.138 ±0.149	0.240 ±0.084	0.104 ±0.091	0.147 ±0.109	0.187 ±0.060	0.159 ±0.101	
GPR	×	0.000 ±0.000	0.147 ±0.202	0.130 ±0.186	0.213 ±0.307	0.124 ±0.170	0.117 ±0.162	0.227 ±0.352
	✓	0.000 ±0.000	0.180 ±0.249	0.050 ±0.112	0.124 ±0.170	0.137 ±0.192	0.089 ±0.122	
GVAUX	×	0.115 ±0.115	0.151 ±0.099	0.219 ±0.133	0.240 ±0.121	0.306 ±0.121	0.379 ±0.123	0.359 ±0.279
	✓	0.000 ±0.000	0.153 ±0.143	0.226 ±0.222	0.109 ±0.114	0.225 ±0.165	0.245 ±0.080	
GVM	×	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.080 ±0.179	0.067 ±0.149	0.000 ±0.000	
	✓	0.100 ±0.224	0.000 ±0.000	0.100 ±0.224	0.100 ±0.224	0.000 ±0.000	0.000 ±0.000	
GVN	×	0.033 ±0.075	0.219 ±0.312	0.167 ±0.236	0.228 ±0.221	0.212 ±0.329	0.083 ±0.118	0.160 ±0.358
	✓	0.031 ±0.069	0.000 ±0.000	0.176 ±0.258	0.200 ±0.278	0.142 ±0.195	0.036 ±0.081	
GVNF	×	0.000 ±0.000	0.080 ±0.179	0.180 ±0.249	0.227 ±0.352	0.260 ±0.241	0.260 ±0.241	0.160 ±0.358
	✓	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.147 ±0.202	0.000 ±0.000	
GVT	×	0.062 ±0.061	0.142 ±0.156	0.120 ±0.113	0.155 ±0.046	0.174 ±0.096	0.117 ±0.083	0.161 ±0.162
	✓	0.081 ±0.102	0.131 ±0.143	0.050 ±0.112	0.056 ±0.082	0.108 ±0.066	0.150 ±0.112	
GWC	×	0.000 ±0.000	0.000 ±0.000	0.067 ±0.149	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	
LP	✓	0.000 ±0.000	0.050 ±0.112	0.000 ±0.000	0.044 ±0.099	0.040 ±0.089	0.000 ±0.000	
LSADJ	×	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.200 ±0.447	0.000 ±0.000	
	✓	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.050 ±0.112	0.000 ±0.000	
LSADV	×	0.000 ±0.000	0.080 ±0.179	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.067 ±0.149
LSN	×	0.000 ±0.000	0.067 ±0.149	0.044 ±0.099	0.137 ±0.192	0.000 ±0.000	0.050 ±0.112	
	✓	0.000 ±0.000	0.100 ±0.224	0.000 ±0.000	0.040 ±0.089	0.044 ±0.099	0.134 ±0.128	
LSPR	×	0.193 ±0.124	0.186 ±0.077	0.274 ±0.118	0.286 ±0.036	0.214 ±0.067	0.268 ±0.155	
	✓	0.000 ±0.000	0.323 ±0.111	0.216 ±0.094	0.197 ±0.129	0.165 ±0.105	0.201 ±0.084	
LSV	×	0.000 ±0.000	0.106 ±0.148	0.180 ±0.249	0.146 ±0.182	0.170 ±0.122	0.153 ±0.166	0.029 ±0.064
	✓	0.000 ±0.000	0.050 ±0.112	0.000 ±0.000	0.062 ±0.138	0.145 ±0.149	0.088 ±0.136	
LWCO	×	0.000 ±0.000	0.000 ±0.000	0.036 ±0.081	0.024 ±0.053	0.031 ±0.069	0.082 ±0.126	0.073 ±0.163
	✓	0.000 ±0.000	0.000 ±0.000	0.123 ±0.116	0.000 ±0.000	0.036 ±0.081	0.000 ±0.000	
QC	×	0.000 ±0.000	0.200 ±0.447	0.200 ±0.447	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	
QM	×	0.277 ±0.171	0.196 ±0.162	0.235 ±0.156	0.288 ±0.091	0.232 ±0.097	0.163 ±0.107	0.237 ±0.160
	✓	0.067 ±0.092	0.224 ±0.062	0.216 ±0.152	0.364 ±0.190	0.373 ±0.107	0.224 ±0.152	
WM	×	0.067 ±0.149	0.183 ±0.171	0.374 ±0.172	0.133 ±0.183	0.359 ±0.330	0.564 ±0.178	0.288 ±0.287
	✓	0.100 ±0.224	0.141 ±0.199	0.337 ±0.208	0.258 ±0.280	0.436 ±0.185	0.200 ±0.189	
WO	×	0.000 ±0.000	0.036 ±0.081	0.000 ±0.000	0.086 ±0.081	0.031 ±0.069	0.000 ±0.000	0.040 ±0.089
	✓	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.057 ±0.128	0.031 ±0.069	0.000 ±0.000	
WR	×	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.040 ±0.089	0.044 ±0.099	0.025 ±0.056
	✓	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.057 ±0.078	0.000 ±0.000	
XADJPR	×	0.000 ±0.000	0.000 ±0.000	0.200 ±0.447	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	
XNUC	×	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.100 ±0.224	0.200 ±0.447
XVCO	×	0.000 ±0.000	0.067 ±0.149	0.050 ±0.112	0.073 ±0.104	0.036 ±0.081	0.057 ±0.128	0.044 ±0.099
	✓	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.031 ±0.069	0.040 ±0.089	0.040 ±0.089	

Table 10
Micro-averaged F_1 results per category for *Llama-3.1-8B-Instruct* in the coarse-grained setting, using varying amounts of randomly-sampled pairs of ICL examples.

		Llama-3.1-8B-Instruct						*70B*
	Tags	**0**	**2**	**4**	**6**	**8**	**10**	**6**
Code-switching	×	0.000 ±0.000	0.000 ±0.000	0.050 ±0.112	0.050 ±0.112	0.073 ±0.163	0.233 ±0.325	
	✓	0.000 ±0.000	0.040 ±0.089	0.089 ±0.122	0.197 ±0.192	0.194 ±0.211	0.292 ±0.443	0.175 ±0.186
DMC	×	0.833 ±0.032	0.807 ±0.058	0.813 ±0.059	0.814 ±0.064	0.810 ±0.064	0.800 ±0.052	
	✓	0.784 ±0.058	0.826 ±0.056	0.826 ±0.052	0.827 ±0.051	0.818 ±0.064	0.832 ±0.088	0.854 ±0.036
Form	×	0.380 ±0.140	0.447 ±0.123	0.534 ±0.047	0.529 ±0.117	0.470 ±0.125	0.496 ±0.100	
	✓	0.405 ±0.130	0.477 ±0.049	0.413 ±0.147	0.497 ±0.123	0.488 ±0.118	0.453 ±0.118	0.551 ±0.090
Grammar	×	0.203 ±0.047	0.241 ±0.029	0.247 ±0.039	0.268 ±0.058	0.267 ±0.014	0.302 ±0.048	
	✓	0.228 ±0.039	0.251 ±0.035	0.261 ±0.026	0.306 ±0.025	0.284 ±0.066	0.282 ±0.045	0.333 ±0.041
Infelicities	×	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	
	✓	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000	0.000 ±0.000
Lexico-grammar	×	0.029 ±0.064	0.057 ±0.128	0.068 ±0.064	0.059 ±0.086	0.061 ±0.093	0.059 ±0.084	
	✓	0.031 ±0.069	0.044 ±0.099	0.055 ±0.079	0.064 ±0.095	0.092 ±0.061	0.044 ±0.063	0.117 ±0.149
Lexis	×	0.086 ±0.067	0.173 ±0.032	0.157 ±0.054	0.159 ±0.071	0.136 ±0.019	0.143 ±0.051	
	✓	0.091 ±0.050	0.140 ±0.058	0.167 ±0.044	0.168 ±0.076	0.182 ±0.028	0.176 ±0.045	0.201 ±0.051
Punct.	×	0.183 ±0.202	0.155 ±0.168	0.194 ±0.129	0.136 ±0.137	0.089 ±0.085	0.152 ±0.103	
	✓	0.097 ±0.096	0.178 ±0.160	0.181 ±0.149	0.222 ±0.102	0.191 ±0.150	0.156 ±0.209	0.262 ±0.102
Word	×	0.040 ±0.089	0.092 ±0.064	0.081 ±0.063	0.109 ±0.078	0.144 ±0.133	0.051 ±0.071	
	✓	0.000 ±0.000	0.118 ±0.080	0.122 ±0.109	0.066 ±0.067	0.144 ±0.134	0.116 ±0.117	0.117 ±0.129

Table 11
Results per tag for *Llama-3.1-8B-Instruct* and *Llama-3.3-70B-Instruct* in terms of micro-averaged F_1-measure for the context prompt setting, using fine-grained tags. Missing tags indicate the model did not make any predictions for that class. Only non-zero results are shown.

	8B	**Tags**	***70B***	**Tags**
CSINTRA			0.086 ±0.121	×
DMCC	0.594 ±0.120	×	0.722 ±0.098	×
	0.515 ±0.186	✓		
FS	0.217 ±0.030	×	0.485 ±0.187	×
	0.224 ±0.123	✓		
GA			0.109 ±0.073	×
	0.031 ±0.069	✓		
GNN	0.052 ±0.072	×	0.098 ±0.173	×
	0.138 ±0.148	✓		
GPP	0.029 ±0.042	×	0.070 ±0.102	×
	0.036 ±0.052	✓		
GVNF			0.040 ±0.089	×
GVT			0.061 ±0.086	×
LWCO	0.033 ±0.075	×	0.033 ±0.075	×
LSN			0.033 ±0.075	×
LSPR			0.107 ±0.106	×
QM			0.117 ±0.168	×
	0.067 ±0.149	✓		
WM			0.024 ±0.053	×
XVCO			0.144 ±0.221	×