Behind the Smokescreen: Uncovering Cross-LLM Bias and Reasoning Contradictions of LLM-as-a-judge for Generated Feedback

Céline Leuzinger
University of Gothenburg, Sweden
celine.leuzinger@gu.se

Abstract

The field of Feedback Generation for language learners has received growing attention over the last few years, bringing exciting perspectives for students and teachers alike. However, the lack of reliable and efficient evaluation method remains a major challenge. Concurrently, LLM-as-a-judge has appeared as a possible solution to evaluate natural language. Concerns have yet been raised regarding their capabilities in certain tasks, and their tendency to prefer their own output, known as self-bias. The present work tests whether LLMs are capable of assessing feedback similarly to human through G-Evaluation, an LLM-as-a-judge pipeline enhanced with automatic Chain-of-Thought. We also test whether models prefer GPT-generated feedback to human feedback, and control if the reasons provided by the models match the scores they attribute. Our results show that the (dis)agreement patterns of LLMs mirror the (dis)agreement patterns of human raters. Further, there exists a clear preference for GPT-generated texts compared to human texts, even when equal in quality. Finally, LLMs tend to give high scores even when their reasonings are unfavourable, raising further concerns regarding the reliability of LLM-as-a-judge.

1 Introduction

The field of Feedback Generation (FG) for language learners has gained growing attention over the last few years. FG was the object of a shared task in 2022, and numerous studies have since continued to advance the field, making use of powerful Large Language Models (LLMs) to generate explanatory notes for learner errors (Song et al. 2024, Coyne et al. 2025, Ye et al. 2025, Vainikko et al. 2025).

A major challenge in Feedback Generation is the lack of efficient and reliable evaluation method to determine the quality of the generated feedback. Studies have had to turn to either unreliable automatic metric (Nagata et al. 2023, Ye et al. 2025), or costly and time-consuming manual evaluation (Song et al. 2024, Coyne et al. 2025).

However, the advent of a technique commonly dubbed "LLM-as-a-judge" (Gu et al., 2026) has brought hope to break free from the trade-off between quality and efficiency. This technique consists of prompting an LLM to evaluate a piece of text in the form of a score, according to one or several criteria, and has been adopted in various studies (Chen et al. 2023, Chiang and Lee 2023, Han et al. 2025). LLM-as-a-judge has evolved throughout the years, giving rise to more robust and efficient systems (Liu et al., 2023). Studies have also concentrated on explainability, either by dividing the evaluation task into small, explainable steps (Lee et al., 2025) or prompting the LLMs to explain the reason for their scores (Xu et al. 2023, Liu et al. 2023).

While this brings exciting perspectives for the field of Feedback Generation, a number of concerns have also been raised regarding the use of LLMs as evaluators. The performance of LLMs depends heavily on the task at hand (Bavaresco et al., 2025), and they suffer from a variety of biases (Gu et al., 2026), among which self-bias, a tendency through which they prefer their own outputs to those of other models (Spiliopoulou et al., 2025). LLM-as-a-judge pipelines must therefore be carefully tested before they can be applied in Feedback Generation, or any other NLG tasks.

The present paper tests the performance of G-Evaluation in Feedback Generation. We also check whether LLMs prefer GPT-generated texts to human texts, even when not self-generated. We dub this phenomenon "cross-LLM bias". This is to be differentiated from "self-bias", which is only concerned with the LLMs' own outputs. Finally, we also control whether the reasons provided by

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 15th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2026)
pages 269–287, November 19–20, 2026, University of Tartu Library

the LLM, dubbed "reasonings", match the scores attributed.

Our contributions are the following:

1. We show that certain LLMs show high agreement with human raters, but only for majority labels describing high-quality feedback. For minority classes describing low-quality feedback, the agreement is low both among human raters and between human raters and LLMs;

2. We provide evidence that LLMs prefer GPT-generated texts even when not self-generated, a phenomenon we dub "cross-LLM bias";

3. We reveal that reasonings and scores do not always align, with LLMs giving overly high scores even when the reasoning is unfavourable.

The code for our G-Evaluation pipeline is available in our Github repository.[1]

2 Background

In 2022, Nagata et al. (2023) organised the very first shared task on Feedback Generation for language learners. The goal was to create systems capable of generating a short explanatory note given a span containing a preposition error in English. The field of Feedback Generation has continued to gain attention in the following years (Fei et al. 2023, Song et al. 2024, Maity and Deroy 2025, Vainikko et al. 2025), opening up opportunities for both language learners and teachers. However, the field is facing a major challenge: the lack of efficient and reliable evaluation method.

Studies in Feedback Generation have adopted one of two techniques to evaluate the generated feedback: automatic metrics (Nagata et al. 2023, Ye et al. 2025) and manual evaluation (Song et al. 2024, Coyne et al. 2025), both of which have their advantages and drawbacks. Automatic metrics include BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005), and measure textual quality based on n-gram overlap with a reference text. While automatic metrics are convenient and resource-efficient, important concerns have been raised regarding their misalignment with certain tasks and poor correlation with human judgments (Reiter, 2018; Schmidtová et al., 2024). Further, automatic metrics require the creation of a reference dataset, a long and costly process in the case of Feedback Generation.

Manual evaluation entails the recruitment of human experts to assess the quality of the generated feedback based on evaluation criteria. While more reliable than automatic metrics, human evaluation also has its drawbacks: it is a costly and timely procedure that requires extensive expertise, both for the evaluation itself and the definition of evaluation criteria. Coyne et al. (2025) proposed an evaluation grid for generated feedback, based on a set of seven criteria, taking into account the quality of the feedback comment as well as its impact on the learner. While this grid offers a comprehensive framework to evaluate feedback, the problems of time, cost and expertise remain.

These issues, however, could potentially be tackled by a new reference-free, automated evaluation method commonly dubbed "LLM-as-a-judge" (Gu et al., 2026). This technique consists of providing an input and evaluation criteria to an LLM, and prompt it to evaluate the output based on the criteria provided. A large number of studies have shown that LLMs align with human judgments in a range of NLP tasks, such as story generation (Chen et al., 2023; Chiang and Lee, 2023), translation (Xu et al., 2023), as well as dialogue generation and text summarisation (Liu et al., 2023). Several enhanced versions of LLM-as-a-judge have been developed to further improve LLM-based evaluation, among which G-Evaluation (Liu et al., 2023), a pipeline that divides the rating task into evaluation steps using Chain-of-Thought (CoT), and shows higher correlation with human judgments than traditional LLM-as-a-judge. There has also been a push towards explainability, through both task division into small, explainable steps (Lee et al., 2025) and provision of a reasoning to justify the scores provided (Liu et al. 2023, Xu et al. 2023).

Several issues have also been raised regarding the use of LLM-as-a-judge. Performance varies greatly depending on the task at hand and model used (Bavaresco et al., 2025). Further, LLMs suffer from self-bias, also called self-preference or self-enhancement bias, a tendency through which they prefer their own output to those of other models (Gu et al., 2026). Previous studies on the matter have sometimes failed to distinguish self-bias from genuine differences in quality (Spiliopoulou et al., 2025), as self-bias can only be evidenced if the out-

[1] https://github.com/CelineNausicaa/g_eval_generated_feedback

Criterion	Scoring	Defining question
Relevance	Binary	Is the feedback comment relevant to the error?
Factuality	Binary	Is the information in the feedback comment factually correct?
Causality	Binary	Does the feedback comment explain what is wrong and why?
Action.*	Binary	Does the feedback comment explain what to do to fix the error?
Comp.*	Binary	Is the feedback comprehensible to the assumed learner (B1-B2 level)?
Scope	Binary	Does the feedback comment contain unnecessary or out-of-scope content?
Quality	Likert (1-5)	Overall, how good is this feedback comment?

Table 1: Criteria of "good" feedback, along with their scoring type and defining question. A detailed description of each is provided in Appendix B. These criteria are directly taken from Coyne et al. (2025). *"Comp" and "Action." stand for "Comprehensibility" and "Actionability" respectively.

put is compared to a gold standard reference (Chen et al., 2025). Spiliopoulou et al. (2025) provided evidence that LLMs tend to favour both their own outputs and the outputs of models coming from the same family, dubbing this phenomenon "family bias". Wataoka et al. (2025) suggested that the cause of self-bias laid in perplexity, with LLMs favouring texts that display lower perplexity, regardless of the source. Lastly, LLM-as-a-judge pipelines often use large, proprietary models (Chen et al. 2023, Liu et al. 2023, Chen et al. 2025, Gu et al. 2026). This poses several issues, as such models require a large amount of computational resources, lack transparency, and can be taken out of the market without prior notice.

3 Methodology

To test whether G-Evaluation can be used in FG, we create a complete pipeline that grades feedback comments based on the seven criteria of good feedback (Coyne et al., 2025). To check for cross-LLM bias, we run our system on both human and AI-generated feedback. Finally, we annotate a subset of the reasonings provided by the models to control whether they match the scores attributed.

3.1 Dataset

For our experiments, we use W&I+WCF[2] (Coyne et al., 2025), a dataset containing 1156 feedback comments on learner language errors.

Learner data: the W&I+WCF comprises 197 sentences written by learners of English, as well as their corrections. Each sentence contains one grammatical error, highlighted in an error span. Each error is annotated with three error tags, corresponding to three error typologies: EXPECT (Fei et al., 2023), ERRANT (Bryant et al., 2017) and Coyne et al. (2025)'s own typology, which we label "WCF".

Feedback data: each erroneous sentence is accompanied by six pieces of feedback, obtained through six different experimental settings[3]:

- Human setting (*Human feedback*): human teachers manually provided feedback based on the learner sentence and highlighted error span.
- Template setting (*Human-GPT feedback*): a manually crafted feedback template was selected based on the error type. GPT-4o was then used to fill in slots in the template.
- Tagless setting (*GPT feedback*): GPT-4o was used to provide feedback based solely on the learner sentence and highlighted error span.
- EXPECT setting (*GPT feedback*): GPT-4o was used to provide feedback based on the learner sentence, the highlighted error span, and the EXPECT error tag.
- ERRANT setting (*GPT feedback*): GPT-4o was used to provide feedback based on the learner sentence, the highlighted error span, and the ERRANT error tag.
- WCF setting (*GPT feedback*): GPT-4o was used to provide feedback based on the learner sentence, the highlighted error span, and the WCF error tag.

[2]The creators of the dataset did not name it in the original paper. The name W&I+WCF was suggested by the first author of the original paper through personal communication. It stands for "Write & Improve + Written Corrective Feedback".

[3]See Coyne et al. (2025) for the full description of the dataset creation process.

Each setting, except for the Template system[4], generated one piece of feedback per learner sentence, yielding a total of 1156 feedback comments.

Feedback evaluation data: each of the 1156 feedback comments was rated by two human evaluators out of a pool of four, yielding a total of 2312 rated feedback comments. The annotators reported no significant difference in feedback quality between the four GPT experimental settings (Tagless, EXPECT, ERRANT, WCF). We therefore merge these four settings into one "GPT feedback" setting in the rest of this paper. Our dataset therefore contains a mix of human, human-GPT and GPT feedback.

The evaluators annotated the feedback comments based on the seven criteria of good feedback (Coyne et al., 2025), defined in Table 1. See Appendix A for a complete example datapoint from the W&I+WCF dataset, and Appendix B for a detailed description of each criterion.

3.2 Models

We select a total of six open-weights and open-source models, of different sizes and coming from different families:

LLM	N. of parameters
Gemma 3	4 billions
Gemma 3	12 billions
Llama 3.2	3 billions
Llama 3.1	8 billions
Qwen 2.5	32 billions
Deepseek-r1	32 billions

Table 2: Open-weights and open-source models selected for the experiments.

We do not use proprietary models, as we place emphasis on transparency and reproducibility for our experiments, both of which can be problematic with proprietary LLMs (Section 2). We therefore focus on testing small-to-medium open models, which are easily available to most.

3.3 G-Evaluation pipeline

To build our G-Evaluation pipeline, we use the deepeval package,[5] which allows for the creation of custom metrics based on the user's chosen evaluation criteria. We therefore create seven custom metrics based on the criteria of good feedback defined in Table 1. Our metric definitions also include a detailed bullet point list of when to choose each option for binary metrics ("Use Yes if [...]", "Use No if [...]"), and a rubric score range indication for the Likert scale criterion *Quality*. We use the exact criteria definition as provided by Coyne et al. (2025), so as to preserve equity between human evaluators and LLM evaluators. Only a minor change was made for the criteria *Scope*: the human evaluators of the W&I+WCF used 0 to indicate that the feedback did not contain out-of-scope content and 1 if the feedback did contain out-of-scope content. We reversed this logic in our G-Evaluation pipeline, after noticing that the LLMs struggled to assign a score of 0 if the outcome was positive during pre-experiments. The complete metric definitions are included in Appendix B.

The deepeval package includes a default prompt, which asks the LLM to provide a score and a reasoning for a given criterion, based on an input (the erroneous learner sentence) and actual output (the feedback). The pipeline first automatically generates evaluation steps through CoT based on the descriptions of the criteria. Following the steps generated, it attributes a score between 1 and 10 to the actual output. Rather than directly outputting this score, the G-Evaluation pipeline computes the weighted summation of the probability of each scoring outcome (1-10) to obtain a final discrete score between 0 and 1, in increments of 0.1.

For the binary criteria, we then convert the discrete scores to a binary outcome (True/False), and keep the default threshold of 0.5, meaning that any score at 0.6 or higher is considered a success and returns True. For the *Quality* criterion, scores are scaled to the 1-5 Likert scale.

Temperature is set to 0 and we collect our results after one run of our G-Evaluation pipeline for each model. We do not fine-tune our models nor provide any few-shot annotated examples.

3.4 Evaluation

Measuring agreement: we evaluate the models' performance by measuring the alignment of the G-Evaluation outputs with human judgments. To do so, we use both raw agreement and Krippendorff's alpha. Raw agreement is calculated by dividing the number of times the evaluators agree by the total

[4]Only 171 instances of rated feedback were produced with the Template system, against 197 for the other settings. Manual checks of the data revealed that the template system occasionally generated empty feedback, or that no human rating was provided. These datapoints were therefore removed from the present experiments.

[5]https://deepeval.com/docs/metrics-llm-evals

number of evaluations, while Krippendorff's alpha (Krippendorff, 2011) takes into account agreement by chance.

We compute the raw agreement between the LLM and each of the four human raters, for all seven criteria. We also compute the raw agreement between each pair of human raters, in order to compare the LLM-human agreement to the human-human agreement for each piece of feedback. We then average the agreement across pairs of LLM-human and human-human. The average agreement is presented in Table 3. The non-averaged agreement results are presented in Appendix C. We follow the same methodology for Krippendorff's alpha: we calculate alpha between the models and each of the four raters for all criteria, as well as the alpha between each rater pair. The average alpha is presented in Table 4. The non-averaged alpha is presented in Appendix D.

To deepen our analysis, we check for class prevalence for each binary criterion. Given that our dataset is indeed imbalanced (see section 4.1.1), we carry out further analyses on the minority classes. We calculate the proportion of agreement for minority classes (1) between human raters and (2) between human raters and our G-Eval pipeline, using the formula below. With each pair of brackets containing the labels from a rater pair, Min the minority class label and Maj the majority class label:

$$Agr = \frac{N(Min,Min)}{N(Min,Min)+N(Min,Maj)+N(Maj,Min)}$$

Measuring cross-LLM bias: we base our cross-LLM bias test on the *Quality* criterion, as it illustrates the overall quality of the feedback.

We run a linear mixed-effects model to test for the effect of evaluator type (LLM / human) on the scoring of the criterion *Quality*, for both GPT-generated feedback and human feedback. The results are shown in Table 8. We also visualise the score distribution of each LLM as well as human evaluators in Figure 1 and 2.

Measuring alignment between scores and reasonings: to test whether the scores and reasonings provided by the LLMs align, we manually annotate the same random sample of 15 feedback comments for each LLM. Given that each feedback comment is evaluated across 7 criteria, this yields a total of 105 score-to-reasoning pairs per LLM.

We use sentiment analysis as the basis of our annotations: each reasoning is annotated as either "Positive", "Neutral" or "Negative" based on the set of rules below[6]:

- Positive: the reasoning is positive or mostly positive.
  Example: *"The feedback correctly identifies the error by changing 'for' to 'to', addressing the highlighted span. It explains that a to-infinitive is needed for purpose clauses, which is accurate."*

- Neutral: the reasoning contains both positive and negative comments, in equal quantity.
  Example: *"The feedback provides a direct edit suggestion by changing 'for' to 'to', which aligns with evaluation step 1. It also offers a viable candidate edit, fulfilling step 2. However, it doesn't provide multiple options or address broader issues beyond the specific grammatical change, so steps 3 and 4 are not fully addressed."*

- Negative: the reasoning is negative or mostly negative.
  Example: *"The feedback incorrectly states that 'waiting' is a verb or gerund when it's actually a noun in this context. It also suggests changing to 'wait', which isn't the correct noun form for duration."*

We visualise the distribution of scores for each sentiment in Figure 3.

To obtain a more fine-grained analysis, we also convert the corresponding discrete scores into sentiment, following the set of rules below:

- Score $> 0.6 \rightarrow$ Positive
- $0.4 \leq$ Score $\leq 0.6 \rightarrow$ Neutral
- Score $< 0.4 \rightarrow$ Negative

The reasoning is considered to align with the score if both receive the same sentiment label. We compute the accuracy of alignment and extract the mean and mode score per sentiment. These results are in Appendix E.

4 Results and discussion

The results show the complexity of assessing the performance of G-Evaluation. While the high raw

[6]The examples are retrieved from deepseek-r1:32b outputs.

LLM	Relevance	Factuality	Causality	Action.	Comp.	Scope	Quality
Gemma3:4b	96.38	79.25	98.16	98.64	43.91	95.99	28.92
Gemma3:12b	**99.31**	**92.22**	97.38	**99.92**	95.53	**99.10**	**60.51**
Llama3.2:3b	16.01	46.90	32.72	73.23	73.00	93.03	15.08
Llama3.1:8b	96.23	59.17	**98.08**	93.62	58.12	90.15	32.00
Deepseek-r1:32b	86.89	69.80	71.58	99.31	**97.10**	98.05	52.95
Qwen2.5:32b	89.96	73.88	38.12	94.72	93.47	98.32	24.00
Human raters	98.74	93.46	97.30	99.75	95.36	98.10	45.49

Table 3: Average raw agreement between the LLM and the human raters as well as between human raters only. The detail of agreement between each rater-rater pair and each llm-rater pair is provided in Appendix C. All scores are in percentage.

LLM	Relevance	Factuality	Causality	Action.	Comp.	Scope	Quality
Gemma3:4b	**0.024**	-0.054	-0.008	-0.006	-0.338	-0.020	-0.154
Gemma3:12b	-0.002	**0.057**	**0.019**	N/A	**0.165**	-0.004	-0.114
Llama3.2:3b	-0.717	-0.299	-0.483	-0.150	-0.111	-0.035	-0.427
Llama3.1:8b	0.014	-0.186	-0.009	-0.032	-0.201	-0.017	-0.110
Deepseek-r1:32b	-0.044	-0.0865	-0.133	**-0.002**	0.097	0.027	**-0.021**
Qwen2.5:32b	-0.018	-0.056	-0.420	-0.011	0.117	**0.134**	-0.307
Human raters	-0.004	0.159	N/A	N/A	0.251	-0.008	0.015

Table 4: Average alpha score between the LLM and the human raters as well as between human raters only. The detail of agreement between each rater-rater pair and each llm-rater pair is provided in Appendix D. N/A values suggest little to no variability in the data.

agreement suggests that LLMs evaluate similarly to humans, the low alpha scores show that the pipeline struggles with lower-quality feedback. The analyses of cross-LLM bias and alignment between score and reasoning also depict a very contrasted picture.

4.1 Agreement analysis

The results presented in Table 3 show that Gemma3:12b is seemingly the best performing model, with a raw agreement above 90% across all binary metrics. For the Likert scale criterion *Quality*, Gemma3:12b even obtains a higher average agreement with humans than human raters do with each other.

Other models exhibit important differences depending on the criterion, with higher performances recorded for the binary criteria than for the Likert scale criterion. All models obtain a high raw agreement in *Actionability* and *Scope*, while all other criteria show important variation between models. For example, Gemma3:4b underperforms in *Comprehensibility*, and Qwen2.5:32b in *Causality*, but both show high agreement for all other binary criteria. *Quality* is the criterion that shows the lowest agreement, both with LLMs and human raters.

Interestingly, the agreement patterns of most LLMs seem to mirror the agreement patterns of humans: when the Inter-Annotator Agreement (IAA) is high between humans, it is also high between humans and LLMs. Conversely, IAA drops for the criterion *Quality* between humans, as well as between LLMs and humans. Llama3.2:3b is an exception, showing low performance even when human agreement is high, similarly to Qwen2.5:32b for the criterion *Causality*.

The results for Krippendorff's alpha, presented in Table 4, tell a drastically different story: while the high raw agreement suggests that our G-Eval pipeline performs similarly to human raters for binary criteria, the alpha scores show that such results could be due to pure chance. Given that the raw agreement is high, the low alpha scores does not signify that the raters disagree. Rather, it could be due to important class imbalance, rendering chance-corrected agreement obsolete.

Criteria	Relevance	Factuality	Causality	Action.	Comp.	Scope
Proportion of True	> 95%	> 95%	> 95%	> 95%	> 95%	< 5%
Proportion of False	< 5%	< 5%	< 5%	< 5%	< 5%	> 95%

Table 5: Prevalence of each class across the binary criteria, combined for all four human raters.

Criteria	Relevance	Factuality	Causality	Action.	Comp.	Scope
Proportion (%)	0	11.84	6.061	0	12.72	0
Minority class label	False	False	False	False	False	True

Table 6: Proportion of agreement cases, i.e. cases where human annotators agree on the minority class label. For example, a score of 0 for *Relevance* means that the pair of raters never agreed on the minority class label: whenever a rater labeled a feedback as irrelevant (False), the corresponding rater always labeled it as relevant (True). On the other hand, for the criterion *Comprehensibility*, if a rater labeled a feedback comment as incomprehensible (False), the paired rater agreed (also False) in 12.72% of the cases.

4.1.1 Class imbalance

To investigate why the alpha scores are low, we compute the prevalence of each class ("True" / "False") across binary criteria. Table 5 shows a simplified overview of the proportion of each class per criteria across all four raters.

These results show a clear data imbalance, with the label "True" occurring for the vast majority of the data across all criteria. The logic is reversed for the criteria *Scope*, with the label "False" occurring in more than 95% of the cases. This shows that our base dataset is of high quality: human evaluators chose the positive outcome in the vast majority of cases. We therefore carry out further analyses on the rare yet pertinent low-quality feedback (< 5%), to analyse how human evaluators as well as our G-Evaluation pipeline handle such cases.

4.1.2 Minority class agreement

The performance of our G-Evaluation pipeline is evaluated based on whether it mirrors human evaluators. It is therefore of primary importance to first study how human raters handle minority class labels by analysing whether humans agree on such minority cases. High agreement between humans on minority classes would suggest that such cases are rather objective, in the sense that a "ground truth" can be agreed upon. Conversely, low agreement suggests a higher level of subjectivity, where no label is universally accepted. Without such "ground truth" to rely on, evaluating the performance of G-Evaluation based on human labels might reveal itself unfeasible. We calculate the agreement for minority classes between human raters using the formula presented in Section 3.4. The results are presented in Table 6.

Interestingly, the results are low across all criteria. The criteria *Relevance*, *Actionability* and *Scope* even show 0% agreement for their minority classes, meaning that the rater pair never agreed. The numbers are slightly higher for the criteria *Factuality* and *Comprehensibility*, but remain low overall. Together with the raw agreement results from Section 4.1, these scores reveal an interesting insight about the task of feedback evaluation: humans seem to agree on what constitutes "good" feedback, but disagree on what constitutes "bad" feedback. The minority categories (which are always the ones describing lower quality feedback) could therefore be more subjective, with annotators showing different attitudes towards what should be considered flawed or unacceptable feedback.

We also compute the proportions of times when the G-Evaluation pipeline agreed with *at least* one human rater on the minority class label. Results are reported in Table 7, and show that agreement for minority clases is equally low between our G-Evaluation pipeline and human raters. The criteria *Factuality* and *Comprehensibility* show slightly higher agreement than all other criteria, echoing the human agreement results of Table 6.

Once more, the agreement pattern of our G-Evaluation pipeline mirrors the agreement pattern of human raters: there is a lack of agreement on minority classes, which could indicate that evaluating low quality feedback is highly subjective. Given such low agreement, it is impractical to evaluate the performance of our G-Evaluation pipeline based on human raters' answers for the task of feedback

LLM	Relevance	Factuality	Causality	Action.	Comp.	Scope
Gemma3:4b	**2.325**	3.414	0	0	3.351	0
Gemma3:12b	0	4.737	1.613	0	**10.34**	0
Llama3.2:3b	0.614	3.840	1.642	0.162	2.5	0
Llama3.1:8b	1.124	4.069	0	0	3.682	1.716
Deepseek-r1:32b	1.299	5.156	**1.788**	0	8.108	2.174
Qwen2.5:32b	1.680	**5.321**	1.784	**0.813**	8.982	**7.142**
Minority class label	False	False	False	False	False	True

Table 7: Proportion of agreement cases, i.e. cases where our G-Evaluation pipeline agrees with at least one human rater on the minority class label. All results are in percentage.

evaluation. A venue for further research is to test whether raters could come to a consensus with further discussion or clarification of the evaluation guidelines. If disagreements can be solved, then the task of evaluating low-quality feedback could reveal itself more objective than the low minority agreement scores imply.

4.2 Cross-LLM bias

The coefficients in Table 8 show that our LLM evaluators systematically rate human feedback lower than GPT-generated feedback. The p-values are significant for all models except Gemma3:12b. The average score difference between GPT-generated texts and human texts is highest for Gemma3:4b and Llama3.1:8b.

LLM	Coeff.	p-value
Gemma3:4b	-0.939	<0.001
Gemma3:12b	-0.157	0.09
Llama3.1:8b	-0.929	<0.001
Llama3.2:3b	-0.507	<0.001
Qwen2.5:32b	-0.381	<0.001
Deepseek-r1:32b	-0.482	<0.001

Table 8: Results of the linear mixed-effects model for the *Quality* criterion. Coefficients describe the average score difference of *Quality* between the GPT-generated texts and the human texts.

On the other hand, the scores given by human evaluators show little to no difference of quality between GPT-generated feedback and human feedback, as shown in Figure 2.

Figure 1 shows that the score distributions are also strikingly different, both between human and LLM evaluators, and between LLM evaluators themselves. Gemma3:12b, Qwen2.5:32b and Deepseek-r1:32b give mostly high scores of 4 or 5, while the distribution is flatter for all other models.

Previous studies have reported that LLMs tend to prefer their own output (Chen et al., 2025) or output of models from their own family (Spiliopoulou et al., 2025). Our experiments ratify that LLMs favour AI-generated texts even when not self-generated, justifying the name "cross-LLM bias".

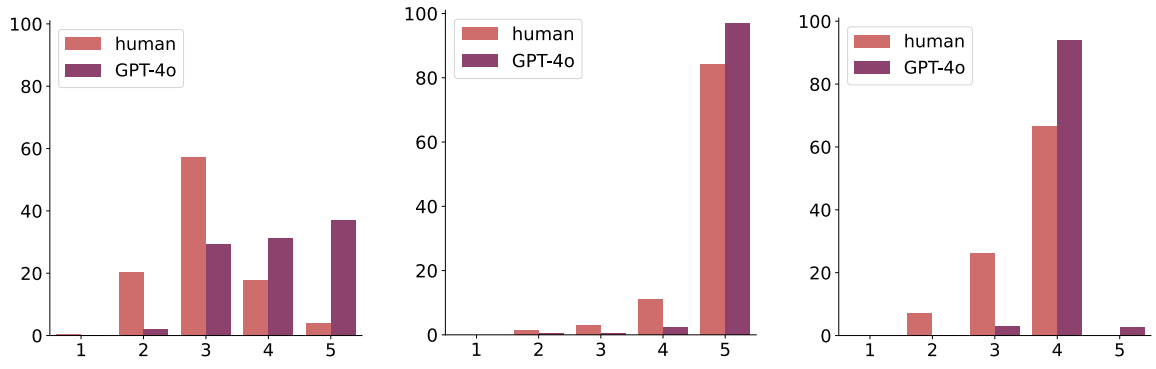
4.3 Alignment between score and reasoning

Figure 3 shows that while models give lower scores when the reasoning provided expresses negative sentiment, the score range remains largely spread out, ranging from 0.0 to 0.7. Only Deepseek-r1-32b displays a slightly more constrained score range. However, it still shows inconsistent scores of up to 0.7. Mode scores for "Negative" reasonings are also high (0.5, 0.6), except for Llama3.1:8b and Qwen2.5:32 (0.3, 0.4).

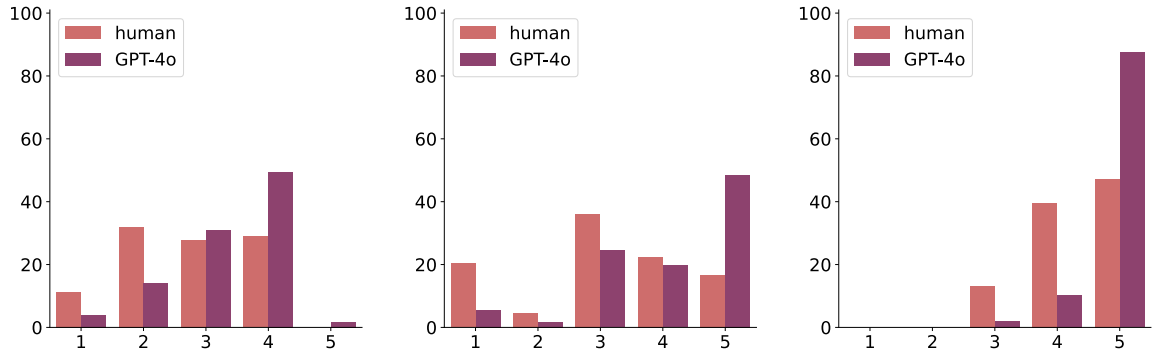
Score range are more constrained for "Neutral" reasonings. However, the scores remain high for most models, spreading from 0.6 to 0.8, with 0.7 and 0.8 being the most common scores. Llama3.2:3b breaks this pattern by displaying an even wider score span, from 0.4 to 0.8. Gemma3:12b also represents a special case for the "Neutral" sentiment, as only one datapoint was annotated as such. We therefore removed it from the results table and plots.

All models show better alignment when the reasoning is "Positive", with mode scores between 0.8 and 1.0, and a score range always higher than the passing threshold (>0.5).

The accuracy of alignment between scores and reasonings varies greatly across models. Gemma3:12b obtains the highest result (98%), despite the mode score for "Negative" sentiment being overly high (0.6). Llama3.2:3b is the lowest



(a) Gemma3:4b (b) Gemma3:12b (c) Qwen2.5:32b



(d) Llama3.2:3b (e) Llama3.1:8b (f) Deepseek-r1:32b

Figure 1: Score distribution for the criterion *Quality* in percentage for each LLM evaluator.

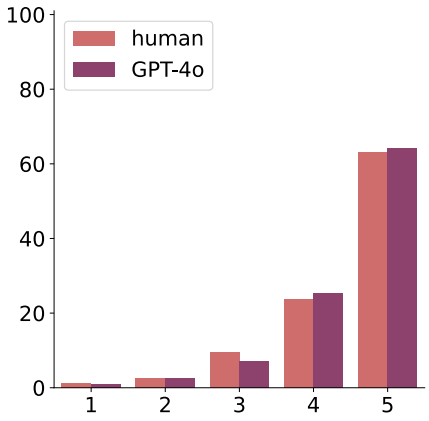


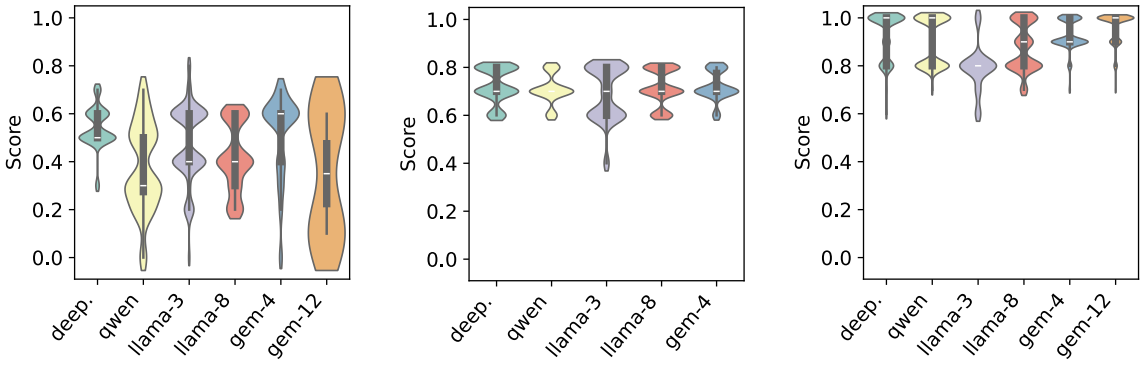
Figure 2: Score distribution for the criterion *Quality* in percentage for human evaluators.

performing model with 53% accuracy, while other models display mixed results (61-83%).

These results suggest that the tested LLMs are capable of attributing high scores to what they consider high quality feedback, but struggle to align their score and reasonings when the feedback is considered lower quality. This is particularly problematic when scores are converted into a binary "True"/"False" label, as even mixed or "Negative" reasonings will give a positive outcome if the threshold is set to the default middle value 0.5. Given that the W&I+WCF contains mostly high quality feedback, the raw agreement results might be inflated by the tendency of LLMs to assign overly high scores. We therefore recommend aligning the threshold based on the scoring pattern of the chosen LLMs. Interestingly, we are facing the same conclusion as Section 4.1: low-quality feedback is more problematic to evaluate than high-quality feedback.

5 Conclusion

In this paper, we show that certain open-source and open-weights LLMs yield results that align with humans in feedback evaluation, but only for majority classes. For minority classes, the agreement between human raters and LLMs (human-LLM) mirrors the agreement among human raters (human-human): evaluators agree on what high-quality feedback is, but disagree on what should be considered low-quality feedback. We also provide evidence that five out of the six LLMs tested suffer from cross-LLM bias and prefer GPT-generated feedback, even when not self-generated. Finally, we expose the tendency of LLMs to attribute high scores even when the reasoning provided is "Neg-



(a) Negative (b) Neutral* (c) Positive

Figure 3: Score distribution per sentiment. For the sake of readability, model names have been abbreviated. deep = Deepseek-r1:32b, qwen = Qwen2.5:32b, llama-3 = Llama3.2:3b, llama-8 = Llama3.1:8b, gem-4 = Gemma3:4b, gem-12 = Gemma3:12b. *Only one datapoint was annotated as "Neutral" for Gemma3:12b over a total of 105. It was therefore removed from the figure.

ative" or "Neutral", providing evidence that there are misalignments between scores and reasonings.

These findings show the complexity of assessing LLM-as-a-judge pipelines as well as the numerous risks of using them. We therefore recommend using G-Evaluation pipelines with great care: pre-experiments should be run to select a model that displays high agreement with human raters across both majority and minority labels. Researchers should also carefully consider the nature of their task, and refrain from using G-Evaluation if the task is highly subjective, as the evaluation of the LLM output becomes impractical without a ground truth. Further, the selected LLMs must show consistency between scores and reasonings. The scoring behaviour of the models must also be tested beforehand, so as to select an appropriate threshold for success. The problem of cross-LLM-bias should also be addressed before LLMs can reliably be used as evaluators, particularly when comparing both human and AI-generated feedback.

In future work, we plan on testing whether some of the disagreements between human raters can be resolved for minority classes, and reflect on whether feedback evaluation can really be "objective". We will also test whether training the models on a subset of annotated feedback comments reduces cross-LLM bias. Finally, we plan on testing how consistent the models are across several rounds of evaluation.

Limitations

While our agreement analysis did account for minority classes, the imbalance of data can still be an issue, as some of the proportions calculated are based on a small number of datapoints. Another limitation of the present paper is that the data for the sentiment analysis was annotated by only one person. Given that sentiment is subjective, the results might vary with a different annotator. Further, the dataset for the sentiment analysis is small, 105 datapoints per LLM.

Ethical considerations

The use of LLM-as-a-judge raises ethical questions that must be discussed. While LLMs can show high raw agreement with human evaluators, their rating behaviour differs widely from humans, due among others to cross-LLM bias and misalignment between scores and reasonings. This could cause important issues if LLMs were to be used as judges in real-world settings, for example to evaluate human teachers or determine the outcome of a language exam for learners. Our G-Evaluation pipeline is built solely for the sake of research and is not meant to be used in any real-world setting.

Acknowledgments

The author of the present paper would like to thank Steven Coyne for his valuable insights and Elena Volodina for her constructive feedback. This work has been supported by Språkbanken - jointly funded by its 10 partner institutions and the Swedish Research Council (2025-2028; project id 2023-00161).

AI Disclosure

☒ Not applicable (no AI tools used)
☐ Assistive use
☐ Editing
☐ Generation of minor elements
☐ Translation
☐ Compression/expansion
☐ Generation from prompt

Details: No AI tools were used in preparing this work.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72. Association for Computational Linguistics.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, Andre Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2025. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–255. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805. Association for Computational Linguistics.

Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. 2023. Exploring the use of large language models for reference-free text quality evaluation: An empirical study. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 361–374. Association for Computational Linguistics.

Zhi-Yuan Chen, Hao Wang, Xinyu Zhang, Enrui Hu, and Yankai Lin. 2025. Beyond the surface: Measuring self-preference in LLM judgments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1653–1672. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631. Association for Computational Linguistics.

Steven Coyne, Diana Galvan-Sosa, Ryan Spring, Camélia Guerraoui, Michael Zock, Keisuke Sakaguchi, and Kentaro Inui. 2025. Annotating errors in english learners' written language production: Advancing automated written feedback systems. In Alexandra I. Cristea, Erin Walker, Yu Lu, Olga C. Santos, and Seiji Isotani, editors, *Artificial Intelligence in Education*, volume 15880, pages 292–306. Springer Nature Switzerland. Series Title: Lecture Notes in Computer Science.

Yuejiao Fei, Leyang Cui, Sen Yang, Wai Lam, Zhenzhong Lan, and Shuming Shi. 2023. Enhancing grammatical error correction systems with explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7489–7501. Association for Computational Linguistics.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Zhouchi Lin, Bowen Zhang, Lionel Ni, Wen Gao, Yuanzhuo Wang, and Jian Guo. 2026. A survey on LLM-as-a-judge. *The Innovation*, 7(6):101253.

Steve Han, Gilberto Titericz Junior, Tom Balough, and Wenfei Zhou. 2025. Judge's verdict: A comprehensive analysis of LLM judge capability through human agreement. (arXiv:2510.09738).

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability. Technical report, University of Pennsylvania.

Yukyung Lee, JoongHoon Kim, Jaehee Kim, Hyowon Cho, Jaewook Kang, Pilsung Kang, and Najoung Kim. 2025. CheckEval: A reliable LLM-as-a-judge framework for evaluating text generation using checklists. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15771–15798. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522. Association for Computational Linguistics.

Subhankar Maity and Aniket Deroy. 2025. Leveraging prompt-tuning for bengali grammatical error explanation using large language models. (arXiv:2504.05642).

Ryo Nagata, Masato Hagiwara, Kazuaki Hanawa, and Masato Mita. 2023. A report on FCG GenChal 2022: Shared task on feedback comment generation for language learners. In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 45–52. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Ehud Reiter. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Patrícia Schmidtová, Saad Mahamood, Simone Balloccu, Ondřej Dušek, Albert Gatt, Dimitra Gkatzia, David M. Howcroft, Ondřej Plátek, and Adarsa Sivaprasad. 2024. Automatic metrics in natural language generation: A survey of current evaluation practices.

Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer. 2024. GEE! grammar error explanation with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 754–781. Association for Computational Linguistics.

Evangelia Spiliopoulou, Riccardo Fogliato, Hanna Burnsky, Tamer Soliman, Jie Ma, Graham Horwood, and Miguel Ballesteros. 2025. Play favorites: A statistical method to measure self-bias in LLM-as-a-judge. (arXiv:2508.06709).

Martin Vainikko, Taavi Kamarik, Karina Kert, Krista Liin, Silvia Maine, Kais Allkivi, Annekatrin Kaivapalu, and Mark Fishel. 2025. Paragraph-level error correction and explanation generation: Case study for estonian. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 953–967. Association for Computational Linguistics.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2025. Self-preference bias in LLM-as-a-judge.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. 2023. INSTRUCTSCORE: Towards explainable text generation evaluation with automatic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5967–5994. Association for Computational Linguistics.

Jingheng Ye, Shang Qin, Yinghui Li, Xuxin Cheng, Libo Qin, Hai-Tao Zheng, Ying Shen, Peng Xing, Zishan Xu, Guo Cheng, and Wenhao Jiang. 2025. EXCGEC: A benchmark for edit-wise explainable chinese grammatical error correction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(24):25678–25686.

A Example datapoint from W&I+WCF

Example datapoint	
Learner sentence	It *became* a big problem now.
Correction	It has become a big problem now.
EXPECT tag	Verb tense
ERRANT tag	R:VERB:TENSE
WCF tag	Tense: Perfect
Feedback setting	Tagless (*GPT-feedback*)
Feedback	The verb 'became' is in the past tense, but the word 'now' indicates the present. To match the present time, the present perfect tense 'has become' is needed. Change 'became' to 'has become' to correctly match the present time indicated by 'now'.
Rater ID	1
Relevance	True
Factuality	True
Causality	True
Actionability	True
Comprehensibility	True
Scope	False
Quality	4

Table 9: Example datapoint from the W&I+WCF dataset.

B Criteria description

Criterion	Definition
Relevance	Is the feedback relevant to the error? A feedback comment is relevant to the error if it attempts to address the issue represented by the highlighted spans in the Source and Corrected sentences. A feedback comment can refer to words outside the highlight, such as to establish context or a reason to make a correction. However, the "error" that is focused on should be within the highlight. Note that a feedback comment can be "relevant" but incorrect or otherwise flawed. Use Yes If: The feedback identifies an error in the highlighted span. The feedback makes some attempt to describe or address the error. Use No If: The feedback comment seems to be addressing a different error outside of the highlight. The feedback comment is nonsensical or does not seem to apply to the sentence. The feedback comment does not talk about a language error, but is focused on something else like content or praising the learner. The feedback seems like a response to the text instead of an instructor's constructive writing feedback (this would also impact the "Unnecessary Content" annotation).
Factuality	Is the information in the feedback comment factually correct? A feedback comment is factual if there are no inaccurate statements about the error, the possible correction(s), or any rule of English mentioned in the feedback text. Use Yes If: Statements referring to the source sentence, such as "you are using A with B," are accurate. The error described by the feedback is indeed an error. The English language rules mentioned in the feedback are valid. Statements like 'must' or "usually" have the correct scope. Use No If: The feedback asserts a rule that is not real or valid. The feedback overgeneralizes a rule of English.
Causality	Does the feedback comment explain what is wrong and why? A feedback comment "explains what is wrong and why" if it successfully identifies the error and provides at least a simple explanation for it. This may or may not involve a grammatical keyword like 'past tense.' A simple example of feedback that does not explain the error is a grammar-check system that provides only underlines and click-to-fix replacements for complex changes. Learners can observe the suggestions and learn, but there is no explicit explanation.The necessary complexity of an explanation varies with error type. Simple errors like spelling can be "explained" with a basic line like "This word may be misspelled". Use Yes If: The feedback identifies a specific grammar point that was attempted unsuccessfully. The feedback provides a pattern that describes the right way to form a construction. The feedback establishes that a certain combination of words is not natural, ideally with a natural alternative or reference to a certain category of words that do or do not fit. Use No If: The feedback consists only of an edit suggestion like "replace A with B," "remove this word," or "add a comma here." The feedback is very vague or open-ended, such as "This is wrong."

Table 10: Part 1: binary criteria of good feedback, along with their defining question and detailed description. Based on Coyne et al. (2025).

Criterion	Definition
Action.	Does the feedback comment explain what to do to fix the error? A feedback comment "explains what to do" if it provides either a concrete edit suggestion or a hint related to the error. Note that a later field will ask which approach was used. Use Yes If: The feedback provides a direct word replacement, deletion, or insertion, such as "add "to" to make this a to-infinitive" or "Remove the extra comma." The feedback provides one viable candidate edit of many, e.g., "replace "bigly" with another adverb such as greatly." The feedback makes a general suggestion that fixes a broad problem, such as "add a verb to complete this sentence." The feedback outlines a grammatical change but does not go into specifics, e.g., "change this verb to the past progressive." Use No If: The feedback describes an error but does not communicate how to fix it. The feedback gives a suggestion that is very vague, such as 'try again.' Note that a broad error may require a broad suggestion, but it should at least be based on the type of error in the sentence. The feedback tells the learner to consult external resources like a dictionary. This is one form of "what to do," but does not explain how to fix the error.
Comp.	Is this feedback comprehensible to the assumed learner? This annotation should reflect your judgement on whether the "assumed learner" - a CEFR B1 or B2 learner who has used academic resources, is likely to understand the comment if they try to read it. These learners can be assumed to know a certain amount of metalinguistic words, and may even be interested in keywords related to an error they made, but there is a limit to how complex a feedback comment can be before it becomes too difficult to be effective. Learners may be intimidated by a long and seemingly difficult comment and not even read it. The best way to measure this is to ask students about feedback they received on their own writing. However, it is not practical in our current research. We are thus asking teachers whether the comment seems reasonably comprehensible for this kind of learner. Use Yes If: The feedback uses a relatively simple way to explain the error and/or correction, given the options. The feedback is not overwhelmingly long or complex. Use No If: The feedback uses linguistic or grammatical terms that even an academic language learner would not know, or uses a large number of terms that would be overwhelming. The feedback has unnecessarily high length or syntactic complexity.
Scope	Does the feedback comment contain unnecessary or out-of-scope content? A good feedback comment is relatively minimal, providing guidance without extraneous information. Use this annotation to flag comments that have unnecessary, questionable, or out-of-scope content. Use Yes If: All the content in the feedback pertains to the error, the way to resolve it, or the concept the learner should learn or review. Use No If: The feedback has something that only makes sense in an AI chat, like 'Let me know if I can help you with anything else!' The feedback contains an assumption about the learner or task, such as nationality, first language, real-world contexts like a class, etc. The feedback is about something other than language correctness, such as the quality of a story or the strength of an argument. The feedback is somehow rude or offensive. The feedback contains praise for the learner for trying hard or for correct text elsewhere. Praise makes sense in many feedback contexts, but not the setting we are building. Please cooperate and flag praise in feedback text.

Table 11: Part 2: binary criteria of good feedback, along with their defining question and detailed description. Based on Coyne et al. (2025). "Comp." and "Action." stand for "Comprehensibility" and "Actionability" respectively.

Criterion	Definition
Quality	Overall, how good is this feedback comment? Overall Quality is a 1-5 measure of a rater's impression of the feedback comment. We place it last so that all prior questions should be answered first to help inform this judgement. Nevertheless, this field is not a direct reflection of prior questions, and there is no formula for deducting points for each "undesirable" answer above it. The score should be attributed following the below guidelines: 1/5: Irrelevant or useless comment. 2-3/5: Potentially valid but flawed comment. 4-5/5: The comment is likely to help the learner if interacted with.

Table 12: Criterion of good feedback *Quality*, along with its defining question and detailed description. Based on Coyne et al. (2025).

C Raw agreement scores per rater

LLM	Relevance	Factuality	Causality	Action.	Comp.	Scope	Quality
Gemma3:4b	0.942	0.787	0.996	0.978	0.458	0.947	0.236
Gemma3:12b	0.985	0.885	0.988	1.000	0.928	0.982	0.981
Llama3.2:3b	0.160	0.480	0.293	0.740	0.723	0.918	0.115
Llama3.1:8b	0.946	0.599	0.995	0.937	0.589	0.885	0.335
Deepseek-r1:32b	0.845	0.682	0.710	0.993	0.938	0.970	0.578
Qwen2.5:32b	0.867	0.715	0.350	0.945	0.907	0.970	0.170

Table 13: Raw agreement per criterion between the LLMs and rater 1.

LLM	Relevance	Factuality	Causality	Action.	Comp.	Scope	Quality
Gemma3:4b	0.971	0.780	0.990	0.995	0.440	0.962	0.278
Gemma3:12b	0.994	0.948	0.984	1.000	0.971	0.998	0.691
Llama3.2:3b	0.156	0.469	0.335	0.721	0.732	0.932	0.112
Llama3.1:8b	0.977	0.568	0.995	0.946	0.577	0.923	0.335
Deepseek-r1:32b	0.872	0.721	0.745	0.991	0.984	0.982	0.585
Qwen2.5:32b	0.910	0.759	0.363	0.946	0.946	0.991	0.174

Table 14: Raw agreement per criterion between the LLMs and rater 2.

LLM	Relevance	Factuality	Causality	Action.	Comp.	Scope	Quality
Gemma3:4b	0.949	0.796	0.949	0.981	0.454	0.972	0.301
Gemma3:12b	0.998	0.935	0.948	0.997	0.963	0.998	0.853
Llama3.2:3b	0.156	0.470	0.325	0.743	0.742	0.943	0.057
Llama3.1:8b	0.957	0.617	0.953	0.925	0.590	0.892	0.378
Deepseek-r1:32b	0.868	0.695	0.677	0.993	0.982	0.992	0.702
Qwen2.5:32b	0.891	0.735	0.403	0.948	0.942	0.988	0.083

Table 15: Raw agreement per criterion between the LLMs and rater 3.

LLM	Relevance	Factuality	Causality	Action.	Comp.	Scope	Quality
Gemma3:4b	0.970	0.970	0.980	1.000	0.475	0.950	0.340
Gemma3:12b	0.994	0.921	0.975	1.000	0.959	0.986	0.182
Llama3.2:3b	0.167	0.457	0.356	0.725	0.723	0.928	0.320
Llama3.1:8b	0.969	0.592	0.980	0.937	0.577	0.906	0.232
Deepseek-r1:32b	0.890	0.694	0.732	0.995	0.982	0.978	0.254
Qwen2.5:32b	0.930	0.746	0.408	0.950	0.944	0.984	0.532

Table 16: Raw agreement per criterion between the LLMs and rater 4.

D Krippendorff's alpha per rater

LLM	Relevance	Factuality	Causality	Action.	Comp.	Scope	Quality
Gemma3:4b	0.054	-0.029	-0.001	-0.008	-0.280	-0.028	-0.204
Gemma3:12b	-0.007	-0.006	-0.005	N/A	0.152	-0.008	0.005
Llama3.2:3b	-0.708	-0.237	-0.545	-0.148	-0.046	-0.042	-0.465
Llama3.1:8b	-0.027	-0.133	-0.002	-0.032	-0.134	-0.005	-0.103
Deepseek-r1:32b	-0.042	-0.030	-0.169	-0.003	0.146	-0.014	0.074
Qwen2.5:32b	-0.023	-0.005	-0.480	-0.027	0.173	0.086	-0.337

Table 17: Krippendorff's alpha per criterion between the LLMs and rater 1.

LLM	Relevance	Factuality	Causality	Action.	Comp.	Scope	Quality
Gemma3:4b	-0.014	-0.050	-0.002	-0.002	-0.360	-0.015	-0.093
Gemma3:12b	-0.001	0.095	-0.007	N/A	0.186	0	-0.026
Llama3.2:3b	-0.724	-0.321	-0.497	-0.161	-0.141	-0.034	-0.452
Llama3.1:8b	0.122	-0.230	-0.002	-0.027	-0.236	0.005	-0.064
Deepseek-r1:32b	-0.040	-0.081	-0.145	-0.004	-0.007	-0.008	0.055
Qwen2.5:32b	-0.007	-0.068	-0.466	-0.027	0.091	0.282	-0.298

Table 18: Krippendorff's alpha per criterion between the LLMs and rater 2.

LLM	Relevance	Factuality	Causality	Action.	Comp.	Scope	Quality
Gemma3:4b	-0.019	-0.095	-0.022	-0.010	-0.384	-0.017	-0.360
Gemma3:12b	0	0.060	-0.026	-0.001	0.186	0	0.132
Llama3.2:3b	-0.727	-0.328	-0.448	-0.134	-0.111	-0.028	-0.682
Llama3.1:8b	-0.021	-0.212	-0.023	-0.038	-0.200	-0.056	-0.247
Deepseek-r1:32b	-0.069	0.120	-0.127	-0.003	0.258	-0.003	0.056
Qwen2.5:32b	-0.056	-0.094	-0.348	0.035	0.117	-0.005	-0.625

Table 19: Krippendorff's alpha per criterion between the LLMs and rater 3.

LLM	Relevance	Factuality	Causality	Action.	Comp.	Scope	Quality
Gemma3:4b	0.078	-0.042	-0.009	-0.004	-0.330	-0.018	0.040
Gemma3:12b	-0.002	0.079	0.113	N/A	0.128	-0.006	-0.569
Llama3.2:3b	-0.708	-0.309	-0.445	-0.158	-0.147	-0.036	-0.108
Llama3.1:8b	-0.015	-0.170	-0.009	-0.032	-0.236	-0.011	-0.023
Deepseek-r1:32b	-0.025	-0.116	-0.092	-0.018	-0.009	0.133	-0.270
Qwen2.5:32b	0.013	-0.056	-0.386	-0.025	0.086	0.174	0.033

Table 20: Krippendorff's alpha per criterion between the LLMs and rater 4.

E Alignment scores

LLM	Accuracy	Average score			Mode score		
		Negative	Neutral	Positive	Negative	Neutral	Positive
Gemma3:4b	0.75	*0.521*	*0.720*	0.930	*0.6*	*0.7*	0.9
Gemma3:12b	**0.98**	0.350	N/A*	0.963	*0.6*	N/A*	1.0
Llama3.2:3b	0.53	0.477	*0.690*	0.790	*0.6*	*0.8*	0.9
Llama3.1:8b	0.61	0.417	*0.714*	0.874	0.4	*0.7*	0.8
Deepseek-r1:32b	0.67	*0.543*	*0.724*	0.930	*0.5*	*0.7*	1.0
Qwen2.5:32b	0.83	0.375	0.*708*	0.901	0.3	*0.7*	1.0

Table 21: Accuracy of the reasoning alignment, as well as the average and mode score per sentiment. Scores that misalign with the reasonings are marked in italics. *Only one datapoint was annotated as "Neutral" out of 105 for Gemma3:12b. It was therefore removed from the table above.