Reviewing the Reviewer: Elevating Peer Review Quality through LLM-Guided Feedback

Anonymous ACL submission

Abstract

Peer review is central to scientific quality, yet reliance on simple heuristics, namely *lazy thinking* and non-*specific* critiques, has threatened review quality. Prior work frames *lazy thinking* detection as single-label classification and stops at detection, yet review segments often exhibit multiple co-occurring issues, and reviewers benefit more from actionable, guideline-aware feedback than from labels alone. We further show that off-the-shelf LLMs prompted for such feedback frequently rewrite the entire review or address the authors rather than the reviewer, motivating an inference-time approach. We introduce an LLM-driven framework that decomposes reviews into argumentative segments, identifies issues violating ARR guidelines, and generates targeted feedback using issue-specific templates refined by a novel genetic algorithm. In a controlled rewriting study, our feedback reduces guideline violations by up to 92.4%. We also release LAZYREVIEWPLUS, the *first* multi-label dataset of 1,309 sentences annotated for *lazy thinking* and *specificity*.

1 Introduction

Peer review is the cornerstone of scientific quality control, ensuring rigorous evaluation of research (Ware and Mabe, 2015). However, the system faces growing strain, particularly in AI, where paper submissions have surged from 1,678 at NeurIPS 2014 to 17,491 in 2024, a 10.4× increase (Wei et al., 2025). This growth, driven by large language models (LLMs) (Liang et al., 2024) and the *publish-or-perish* culture (van Dalen and Henkens, 2012) , has far outpaced the supply of qualified reviewers, even with mandatory reviewing policies.[1] The result is an unsustainable global workload—over 15 million hours annually—that threatens review quality (Aczel et al., 2021).

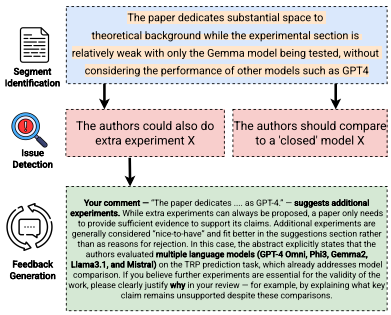


Figure 1: Overall pipeline of our method. We first **identify segments** within a review, then **detect issues**, and finally **generate feedback** to improve each segment.

In NLP research, declining review quality is often attributed to two failure modes flagged by the ACL Rolling Review (ARR) guidelines: *lazy thinking*, where oversimplified heuristics lead reviewers to dismiss submissions (Rogers and Augenstein, 2024), and *specificity issues*, where vague phrasing such as *"X is not clear"* replaces precise feedback such as *"X is not clear because of Y"* (Sadallah et al., 2025).[2] *Lazy thinking* alone accounts for 24.3% of author-reported issues at ACL 2023 (Rogers et al., 2023). For example, the *review segment* (an argumentative unit consisting of one or more sentences) in Fig 1 criticizes a paper for using only GPT-4 and requests additional "nice-to-have" experiments, an example of the ARR guideline issue "authors could also do extra experiment X". We focus on these two issue families because the ARR guidelines define them explicitly and at sufficient granularity to support annotation, detection, and targeted feedback. Our framework is not tied to ARR, as it applies to any venue whose reviewer guidelines specify identifiable issue categories.

Recent efforts to address these issues fall short in two ways. Purkayastha et al. (2025) show that lazy thinking annotations can improve review quality but stop at detection, providing no *verbal feedback*

[1] https://aclrollingreview.org/reviewing-workload-adjustment/

[2] https://aclrollingreview.org/reviewerguidelines

to guide reviewer revision, which is vital for actionable improvement (Jansen et al., 2025). Thakkar et al. (2025) report improved specificity and professionalism in a large-scale ICLR 2024 trial, but their feedback is not aligned with any reviewer guidelines and depends on a closed-source model. A natural alternative is to prompt open-weight LLMs to generate feedback directly. In our pilot study, however, 41% of such outputs extended reviews ('review extension') and 23% misaddressed authors instead of the reviewer ('role separation') (cf. §3.3). These failures underscore the need for inference-time methods that produce specific, targeted, and guideline-aware feedback.

Progress on this problem is further limited by dataset availability. Existing resources annotate only a *single* lazy thinking issue per segment and cover only the "weaknesses" section (Purkayastha et al., 2025), even though multiple issue types can co-occur across sections (cf. Figure 1), and *specificity* is absent entirely.

To address these gaps, we present a framework that produces actionable, guideline-aligned feedback for reviewers using open-weight LLMs. The pipeline segments each review into argumentative units, identifies violations of ARR guidelines, and generates targeted feedback through *issue-specific templates* enriched and refined via a customized *genetic algorithm* (Lee et al., 2025) with verifiable composite rewards tailored to reviewer feedback. Alongside the framework, we release LAZYREVIEWPLUS, a dataset of **1,309** expert-annotated sentences from ARR 2022 and EMNLP 2024 (Dycke et al., 2023) with multi-label annotations for *lazy thinking* and *specificity*.

In summary, our contributions are three-fold: (1) an inference-time *feedback generation* framework combining issue-specific templates and a dedicated *genetic algorithm* with verifiable rewards, improving constructiveness and relevance by about 20% over the strongest baseline and mitigating *review extension* and *role separation* failures of zero-shot LLM feedback; (2) a controlled review-rewriting study with senior PhD reviewers, showing that feedback from our framework reduces guideline violations by up to **92.4**% relative to original reviews and outperforms rewrites guided by issue labels alone; (3) LAZYREVIEWPLUS, the *first* multi-label review dataset annotated for both *lazy thinking* and *specificity*, spanning *weaknesses* and *comments/suggestions* sections across two venues.

2 Construction of LAZYREVIEWPLUS

Building on top of the LAZYREVIEW dataset (Purkayastha et al., 2025), which annotates *500* segments with *16 lazy thinking* classes in single-label form on the *weaknesses* section only, we extend annotations to a single-segment, multi-label form covering both *lazy thinking* and *specificity* (*18* and *7* ARR-defined issues, respectively), spanning both *weaknesses* and *comments/suggestions* sections.[3]

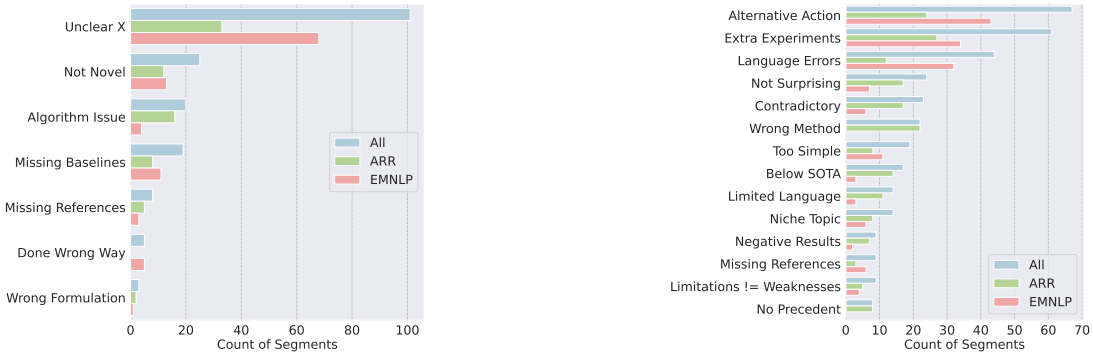
2.1 Data Procurement and Annotation

Underlying Data. To construct LAZYREVIEWPLUS, we sample 100 reviews (50 each) from ARR 2022 and EMNLP 2024 within NLPEER (Dycke et al., 2023), a licensed dataset of ACL and journal reviews (Dycke et al., 2022). We sample from two different timestamps to avoid temporal biases in the dataset creation. Reviews are segmented at the sentence level using SpaCy, covering both *weaknesses* and *comments/suggestions*, unlike LAZYREVIEW which considers only *weaknesses*.[4] This produces 616 and 693 sentences, totaling **1,309** sentences.

Task and Guidelines. We identify review segments with *lazy thinking* and *specificity*. Annotators view the full review along with sentences from the *summary of weaknesses* and *comments, suggestions, and typos* sections. Sentences are labeled using the **B-I-O** scheme (Ramshaw and Marcus, 1999), with each span assigned an issue type. Building on LAZYREVIEW (Purkayastha et al., 2025), we add two categories—*None* (not an issue) and *Not Enough Info* (requires access to paper for labelling)—resulting in **27** issue types (25 from ARR guidelines plus 2 additional classes).[5]

Quality Control. Four Ph.D. students with extensive reviewing experience served as annotators, trained on the finalized guidelines. Disagreements were resolved by a senior student. To ensure reliability, 8% of the dataset (8 reviews; 102 sentences inline with doubly annotating 99 peer review sentences in Purkayastha et al. (2023) and 50 review sentences in Purkayastha et al. (2025)) was double-annotated to compute agreement, while the remaining 92% (92 reviews; 1,207 sentences) was evenly split among annotators. Annotating one review took ∼35 minutes, totaling **180 hours** (∼45 hours per annotator). Inter-Annotator Agreement (Krip-

[3] Full issue lists in Tables 11 and 10; examples in Table 7.
[4] https://spacy.io/api/sentencizer
[5] Final annotation instructions in Appendix §A.2.



(a) *Specificity* Issues

(b) *Lazy Thinking* Issues

Figure 2: *Specificity* issues and overall distribution of *lazy thinking* issues in the LAZYREVIEWPLUS dataset.

pendorff's alpha (Krippendorff, 2004)) was **0.78** for **segment identification** (pairwise 0.74–0.82) and **0.53** for **issue detection** (pairwise 0.51–0.55), consistent with prior peer-review studies (e.g., 0.479 in Kennard et al. (2022) and 0.52 in Purkayastha et al. (2025)) and reflecting the difficulty of identifying *lazy thinking* and *specificity* issues.

2.2 Dataset Analysis

Our dataset, LAZYREVIEWPLUS, contains **1,309** sentences annotated for *lazy thinking* and *specificity* issues (Table 9), comparable to prior human-annotated datasets (e.g., 500 review segments in Purkayastha et al. (2025)). Segment lengths range from 1–25 sentences (mostly single sentences; Fig. 4a) with 1–8 labels per segment (Fig. 4b), totaling **440** segments. For *lazy thinking*, 'authors should do X' and 'Extra Experiments' dominate (Fig. 2b) which is in line with Purkayastha et al. (2025), while *specificity* issues mostly flag clarity problems ('Unclear X'; Fig. 2a), reflecting common non-constructive feedback patterns (Sadallah et al., 2025). These issues — *lazy thinking* and lack of *specificity* — persist across both the ARR 2022 and EMNLP 2024 sections (cf. Figures 2a and 2b), suggesting that reliance on such heuristics remains a recurring trend.

3 Methodology

Our pipeline comprises three stages. First, each review is split into sentences and grouped into argumentative segments via **B-I-O** tagging as described in §3.1.[6] Second, each segment undergoes multi-label **issue detection**, permitting co-occurring labels (e.g., *Extra Experiments* and *Unclear X*) (§3.2). Third, each detected label triggers an issue-specific template in the **feedback generation** module, enriched by a planner and refined via a genetic algorithm, producing multiple targeted feedback pieces for segments with multiple labels (§3.3). The full pipeline is illustrated in Figure 1. We briefly describe the segment identification and issue detection approaches, and primarily focus on our main contribution of **targeted feedback generation**.

3.1 Segment Identification

Task definition. For a review $R = (s_1, \ldots, s_n)$, each sentence s_i is tagged $y_i \in \{B, I, O\}$, where B marks a segment beginning, I an inside sentence, and O one outside any segment. The tagging function $f(s_i) = y_i$ yields $Y = (y_1, \ldots, y_n) = f(R)$.

Approach. A naive approach of prompting an LLM to directly segment a review tends to produce oversized segments that span multiple argumentative units, motivating a finer-grained formulation. Following prior work on LLM-based sequence labeling (Lan et al., 2024; Pichler et al., 2025), we instead prompt the LLM with each review sentence together with the full review as context, and instruct it to assign a 'B/I/O' tag per sentence, detailed in Appendix A.4. The tag definitions match those provided to annotators during dataset construction.

3.2 Issue Detection

Task definition. Given a predefined label set $\mathcal{L} = \ell_1, \ldots, \ell_m$, the goal is to map each review segment $S = (s_a, \ldots, s_b)$ to a subset of labels. Formally, an issue detection function $g(S) \subseteq \mathcal{L}$ produces label sets for all segments in a review, yielding $G(R) = (g(S_1), \ldots, g(S_k))$.

Approach. Following prior work on issue identification (Purkayastha et al., 2025), we first evaluated zero-shot and fine-tuned LLM classification, both of which miss at least 45% of frequent issues on

[6] Sentence split is performed using Spacy

Abstract Feature Questions for Label 'Extra Experiments'

Comment: *"The paper does not explore more complex causal graphs involving potential confounders and mediators."*

Abstract Feature Question (1): Does the reviewer suggest further tests or missing results without justifying why they are necessary?

Figure 3: Example of abstract feature question generated by the LLM for a comment corresponding to the label 'Extra Experiment'.

our dataset (cf. Fig. 10). We attribute this to two factors: issue detection requires intermediate judgments (e.g., whether an experiment request is *nice-to-have* or *must-have*) rather than surface matching, and the input space vastly exceeds the amount of available annotated training data. We therefore decompose the task into two stages: *(1) Abstract Feature Extraction* and *(2) Precision-Oriented Classification*.

1. Abstract Feature Extraction. For each issue class $\ell \in \mathcal{L}$, we prompt an LLM to generate a fixed set of F_c issue-specific Yes/No questions $Q^\ell = \{q_1, \ldots, q_{F_c}\}$ by comparing segments labeled with ℓ and identifying recurring patterns (cf. Fig. 3). For a given segment S_i, the LLM then answers each question with $A_i^\ell \in \{\text{Yes}, \text{No}, \text{Not Relevant}\}$. Responses across all issue classes are aggregated and encoded as a fixed-length feature vector $\mathbf{v}_i \in \{-1, 0, 1\}^{|\mathcal{L}| \times F_c}$, mapping Yes/No/Not Relevant to 1/−1/0. This converts the original reasoning task into pattern recognition over structured features, following prior work on LLM-based feature extraction (Manikandan et al., 2023; Jeong et al., 2025).

2. Precision-Oriented Classification Training classical machine learning models on these structured feature vectors captures non-deterministic correlations with issue labels while remaining effective in low-resource settings without large-scale finetuning (Taye, 2023). To reduce false alarms and improve reviewer trust, we optimize precision-oriented issue detection using the $\mathbf{F_{0.5}}$ score by tuning the hyperparameters on the validation set.

Similar to prior LLM-assisted machine learning approaches (Manikandan et al., 2023; Jeong et al., 2025), our method converts raw text into structured representations, which captures intermediate reasoning steps explicitly and supports precision-oriented classification under limited supervision. Further details are provided in Appendix A.5.

3.3 Feedback Generation

Task definition and challenges. Let each segment be s_i with label l_i and optional context C (e.g., abstract A). The goal is to produce feedback $F_i = g(s_i, l_i \mid C)$; without context, $F_i = g(s_i, l_i)$. For a review $R = (s_1, \ldots, s_n)$ with labels $L = (l_1, \ldots, l_n)$, the feedback sequence is $F = (F_1, \ldots, F_n)$, supporting segment-only and context-informed generation.

Building on prior work on feedback generation (Thakkar et al., 2025), we evaluated zero-shot generation with open-weight LLMs. Many outputs either rewrote/extended entire reviews ("review extension") or incorrectly addressed authors instead of reviewers ("role separation"). In our LAZYREVIEWPLUS dataset, at least 41% of outputs were review extensions and 23% showed role separation. These results motivate an inference-time approach that explicitly mitigates these issues.[7]

Approach. To ensure targeted and diverse reviewer feedback, we adopt a Genetic Evolution-inspired algorithm that iteratively diversifies responses across the detected issues (Lee et al., 2025). Unlike standard zero-shot generation, which frequently produces review extensions or violates reviewer–author role separation, evolutionary search enables controlled exploration and refinement of candidate feedback through mutation and selection. This inference-time optimization encourages issue-focused, reviewer-aligned comments while reducing repetitive or off-target generations. Each piece of feedback is tailored to its corresponding issue, making it more actionable and constructive. The algorithm operates in six stages: *(1) Template Construction*, *(2) Plan Generation*, *(3) Population Initialization*, *(4) Fitness Evaluation*, *(5) Parent Selection and Crossover*, and *(6) Final Candidate Selection*. Stages (4)-(6) are repeated till the maximum number of iterations are reached within the genetic algorithm. Further details are provided in Appendix A.6.

1. Template Construction Unconstrained LLM generation often drifts off-task, producing *review extensions* rather than targeted feedback. Templates provide structural scaffolds that constrain format while leaving the content open to adaptation, mitigating this drift (Kang et al., 2025; Xu et al., 2024). We design **25** author-crafted issue-specific templates $T = \{t_1, \ldots, t_{25}\}$, one per ARR guideline category (Rogers and Augenstein, 2024),

[7] See §A.6.2 for details and examples.

and refine them with senior authors for consistency.[8] Because each template corresponds to a single guideline issue, the set can be re-instantiated for new guidelines or other venues by curating templates against their issue taxonomy; in our setting, drafting all 25 took approximately 250 minutes.

2. Plan Generation To promote diversity in generated feedback, prior work uses model-generated plans to guide unconstrained generation and reduce uncertainty (Narayan et al., 2023; Huot et al., 2023). For each review segment s_i and issue type l_i, we prompt the LLM to produce a plan enriching the template using the paper's abstract, reviewer summary, and noted strengths—mitigating the *role separation* bias. The planner selects relevant knowledge k_i and justifies it with an explanation e_i, forming the plan $\mathcal{P}(r_i, l_i) = \{(k_i, e_i) \mid i = 1, \ldots, N\}$.

3. Population Initialization The LLM is then prompted with the review segment, s_i, the plan, P and the template, t_i to generate n sets of candidates. The initial population set, M_0 is thus given as, $F_0 = \{fb_1, fb_2, \ldots, fb_n\}$.

4. Fitness Evaluation Since we lack ground-truth feedback, we employ composite, verifiable rewards that combine positive scores and penalties to balance generated responses (Hu et al., 2020). We define a fitness function based on intrinsic text properties. Let n_{sent} and n_{words} denote the number of sentences and words. Following (Yaacoub et al., 2025), we reward conciseness: $sc_{len} = \frac{\min(n_{\text{sent}}, n_{\text{max}})}{n_{\text{max}}}$, template adherence via n-gram overlap: $sc_{temp} = \frac{|\text{n-grams}_{\text{fb}} \cap \text{n-grams}_{\text{temp}}|}{|\text{n-grams}_{\text{temp}}|}$, readability via Flesch score (Flesch, 1948): $sc_{read} = \frac{\text{Flesch score}}{100}$, and penalize off-task expressions: $pen_{forb} = \frac{\#\text{forbidden terms}}{n_{\text{words}}}$. The overall fitness is $fit(F_0) = -sc_{len} + sc_{temp} + sc_{read} - pen_{forb}$, enabling self-contained evaluation in LLM-driven *feedback generation*.

5. Parent Selection and Crossover To *select parents*, we use Boltzmann Tournament Selection (Goldberg, 1990), converting fitness scores into probabilities via softmax: $P(fb_i) = \frac{\exp(fit(fb_i)/\tau)}{\sum_{j=1}^{m} \exp(fit(fb_j)/\tau)}$, where τ controls distribution sharpness. Candidates are sampled according to $P(fb_i)$, favoring high-quality feedback while maintaining diversity, selecting exactly $n_{parents}$ for crossover. The *crossover function* combines parent candidates, e.g., fb_1 and fb_2, to generate new offspring: $Cross(fb_1, fb_2) \rightarrow$ new candidates, with the LLM prompted to integrate features from the parents, producing novel and diverse feedback.

6. Final Candidate selection After a fixed number of generations, n_{gen}, we select the candidate with the maximal fitness score, $fb^* = \arg\max_{fb_i \in \mathcal{M}} fit(fb_i)$, where $\mathcal{M}$ is the set of final candidates. If multiple candidates achieve the same maximal score, we perform a cross-over of these candidates to generate a new candidate, which is taken as the final solution.

4 Experimental Setup

Models. We evaluate our pipeline using LLMs from diverse families: Qwen 2.5 7B Instruct (Yang et al., 2025) (Qwen), Yi 1.5 9B Chat (Young et al., 2025) (Yi), Deepseek LLM 7B Chat (Bi et al., 2024) (Deep.), Phi-4 14B (Abdin et al., 2024) (Phi), and GPT OSS 20B (Agarwal et al., 2025) (Oss.). For issue detection, we test various classifiers: KNN, Logistic Regression (L1/L2), Random Forest, Decision Trees, SVM (Linear/RBF/Poly), Gradient Boost, AdaBoost, Extra Trees, and MLP.

Evaluation Metrics. For *feedback generation*, no ground-truth references exist, so we evaluate along four criteria defined in consultation with senior authors: Constructiveness (Fig. 17), Relevance (Fig. 16), Specificity (Fig. 18), and Conciseness (Fig. 15). For automated evaluation, we use `Prometheus-V2` (Kim et al., 2024) to score responses on a 1–5 scale per criterion, following prior work (Sahnan et al., 2025; Maurya et al., 2025); we select it over `Prometheus-V1` and GPT-4o (Hurst et al., 2024) based on the alignment analysis in §A.6.1. For human evaluation, following prior work on peer review (Purkayastha et al., 2025; Lu et al., 2025), three Ph.D. students fluent in English and experienced in peer reviewing rated 100 responses per issue type from our best-performing model (Phi), applying the same criteria as `Prometheus-V2`.

For *segment identification*, we report Precision, Recall, and F1. For *issue detection*, we report macro-averaged $F_{0.5}$ and Precision, in line with the precision-oriented formulation in §3.2. Both are computed against ground-truth annotations in LAZYREVIEWPLUS.

Data Split and Validation. Our *feedback generation* and *segment identification* approaches do not require training, and therefore results are reported on the full dataset. For *issue identification*, we use 90% of the data for training and 10% for hyper-

[8] Full list in Table 12.

parameter tuning, with a 5-fold cross-validation split at the review level.[9]

Baselines. Given a review segment s_i and the issue set $\mathcal{L} = \{l_1, \ldots, l_n\}$, we evaluate the following baselines for *feedback generation*: (i) 1-pass (`1-pass`): the LLM generates feedback from s_i and l_i; (ii) Template-guided (`Temp`): the LLM incorporates a template t_i with s_i and l_i; (iii) Plan-then-generate (`Plan`): the LLM devises a plan $\mathcal{P}i$ for l_i and then integrates it into t_i to generate feedback; (iv) Best-of-N (`BoN`) (Brown et al., 2024): we sample $N = n \times n_{gen}$ candidates and select the best using our fitness function; (v) Self-refinement (`Self-Ref.`) (Madaan et al., 2023): we sample n candidates and refine each over n_{gen} iterations, retaining the final output.[10]

5 Results and Discussion

In this section, we present results across the three subtasks of our pipeline, which together establish the configuration used in the downstream review-rewriting study in §6.

For *segment identification*, following prior work on LLM-based sequence labeling (Lan et al., 2024; Pichler et al., 2025), we cast segment detection as sentence-level **B-I-O** tagging in a zero-shot setting (full prompt and tag definitions in §3.1). Among the five LLMs evaluated, Phi achieves the best performance with Precision, Recall, and F1 scores of **0.81**, **0.77**, and **0.79**, outperforming the next-best model by at least 0.06 F1.[11]

For *issue detection*, following the precision-oriented setup in §3.2, the strongest configuration combines Phi-generated abstract features with an Extra Trees classifier, yielding macro-averaged $F_{0.5}$ of **0.51** and Precision of **0.75** under 5-fold cross-validation, outperforming the strongest baseline (`RoBERTa`) by 0.12 $F_{0.5}$ points. Per-fold variance is below 0.01 (cf. Table 13 and Table 15), and the full ablation, baseline comparisons, and experimental setup are reported in §A.5.

Based on these results, we adopt Phi for both segment identification and abstract feature generation, paired with the Extra Trees classifier for issue detection, throughout the remaining experiments. With the upstream components fixed, the following sections evaluate the central contribution of our framework for *feedback generation*.

[9] Details about the split are provided in §A.5.4

[10] Hyperparameters are listed in Table 27.

[11] The full experimental setup is provided in Sec §A.4

Metric	Model	1-pass	BoN	Self-Ref.	Temp	Plan	Ours
Const.	Yi	1.9	2.0	2.3	2.8	3.0	3.9
Rel.	Yi	2.0	2.2	2.4	2.9	3.1	3.8
Const.	Phi	2.1	2.2	2.4	3.1	3.3	**4.3**
Rel.	Phi	2.2	2.3	2.6	3.2	3.4	**4.3**
Const.	Qwen	1.8	2.0	2.2	2.9	3.1	3.8
Rel.	Qwen	1.9	2.1	2.4	3.0	3.2	3.8
Const.	Deep.	1.7	1.8	2.4	2.7	2.9	3.5
Rel.	Deep.	1.8	1.9	2.4	2.8	3.0	3.6
Const.	Oss.	1.9	2.0	3.0	2.8	3.0	4.0
Rel.	Oss.	2.0	2.1	3.1	2.9	3.1	4.1

Table 1: Model performance on **feedback generation** across six methods for **Constructiveness (Const.)** and **Relevance (Rel.)**. Overall best results are **bolded**, method-wise best are underlined.

Method	Const.	Relev.	Spec.	Conc.
1-pass	1.8	1.9	2.2	2.2
BoN	2.4	2.4	2.5	2.5
Self-Ref.	2.5	2.6	2.6	2.6
Temp	2.8	2.7	2.8	2.8
Plan	3.2	3.1	3.2	3.2
Ours	**4.1**	**4.2**	**4.0**	**4.3**

Table 2: Human evaluation of the best-performing model (Phi) on the **feedback generation** task, reporting Constructiveness (Const.), Relevance (Relev.), Specificity (Spec.) and Conciseness (Conc.)

Variant	Const.	Rel.
Full Algorithm (Ours)	**4.3**	**4.3**
w/o. Template Construction	3.6	3.5
w/o. Plan Generation	3.8	3.8
w/o. Population Initialization	3.9	3.9
w/o. Fitness Evaluation	3.7	3.7
w/o. Parent Selection & Crossover	3.9	3.8
w/o. Final Candidate Selection	3.8	3.7

Table 3: Ablation study of the various components for the best performing model, Phi within the genetic algorithm focusing on the automated metrics.

5.1 Effectiveness of feedback generation

We present the results of our **automated evaluation** in Table 1.[12] We find that our approach consistently outperforms all baselines, producing targeted, diverse feedback with the same number of candidates. While Best-of-N (`BoN`) boosts diversity, it lacks structure; Self-refinement (`Self-refine`) improves quality but still drifts off-task without external rewards (Lee et al., 2025).

We also find that template-based prompting (`Temp`) consistently boosts performance over zero-shot generation: Phi improves from 2.1 to 3.1 in *Constructiveness* and from 2.2 to 3.2 in *Relevance* (Xu et al., 2024; Kang et al., 2025). The

[12] Full results in Table 16

Model	Variant	Const.	Relev.
Yi	Full Fitness Function	**3.9**	**3.8**
	w/o Length Score (sc_{len})	3.6	3.6
	w/o Template Adherence (sc_{temp})	3.7	3.7
	w/o Readability (sc_{read})	3.8	3.8
Qwen	Full Fitness Function	**3.8**	**3.8**
	w/o Length Score (sc_{len})	3.5	3.5
	w/o Template Adherence (sc_{temp})	3.6	3.6
	w/o Readability (sc_{read})	3.7	3.7
DeepSeek	Full Fitness Function	**3.5**	**3.6**
	w/o Length Score (sc_{len})	3.2	3.3
	w/o Template Adherence (sc_{temp})	3.3	3.4
	w/o Readability (sc_{read})	3.4	3.5
Oss.	Full Fitness Function	**4.0**	**4.1**
	w/o Length Score (sc_{len})	3.7	3.6
	w/o Template Adherence (sc_{temp})	3.8	3.7
	w/o Readability (sc_{read})	3.9	4.0
Phi	Full Fitness Function	**4.3**	**4.3**
	w/o Length Score (sc_{len})	3.8	3.7
	w/o Template Adherence (sc_{temp})	3.7	3.6
	w/o Readability (sc_{read})	4.1	4.0
	w/o Forbidden Term Penalty (pen_{forb})	4.2	4.2

Table 4: Ablation study on the various rewards within the fitness function in the feedback generation approach, reporting Constructiveness (Const.) and Relevance (Relev.) across models.

Model	Variant	Const.	Relev.
Yi	Ours (Abstract + Summary + Strengths)	**3.9**	**3.8**
	w/o Abstract	3.6	3.7
	w/o Summary	3.7	3.7
	w/o Strengths	3.8	3.8
	w/o Abstract + Summary	3.5	3.6
	w/o Abstract + Strengths	3.6	3.7
Qwen	Ours (Abstract + Summary + Strengths)	**3.8**	**3.8**
	w/o Abstract	3.5	3.5
	w/o Summary	3.6	3.6
	w/o Strengths	3.7	3.7
	w/o Abstract + Summary	3.4	3.5
	w/o Abstract + Strengths	3.5	3.6
DeepSeek	Ours (Abstract + Summary + Strengths)	**3.5**	**3.6**
	w/o Abstract	3.2	3.3
	w/o Summary	3.3	3.4
	w/o Strengths	3.4	3.5
	w/o Abstract + Summary	3.1	3.2
	w/o Abstract + Strengths	3.2	3.3
Oss.	Ours (Abstract + Summary + Strengths)	**4.0**	**4.1**
	w/o Abstract	3.7	3.8
	w/o Summary	3.8	3.9
	w/o Strengths	3.9	4.0
	w/o Abstract + Summary	3.6	3.8
	w/o Abstract + Strengths	3.7	3.9
Phi	Ours (Abstract + Summary + Strengths)	**4.1**	**4.2**
	w/o Abstract	3.8	4.0
	w/o Summary	3.9	4.0
	w/o Strengths	4.0	4.1
	w/o Abstract + Summary	3.6	3.9
	w/o Abstract + Strengths	3.7	3.9
	w/o Summary + Strengths	3.7	3.9

Table 5: Planner ablation study across Constructiveness (Const.) and Relevance (Relev.) on feedback generation for all models.

Plan-then-Generate paradigm yields more modest additional gains (e.g., *Constructiveness* 3.1 $\rightarrow$ 3.3), suggesting that structured planning contributes but that template scaffolding drives most of the improvement. Across models, Phi is the strongest performer, consistent with its reported strength on reasoning-oriented tasks (Abdin et al., 2024).

For **human evaluation** on the best-performing model (Phi), we obtain substantial inter-annotator agreement across all four metrics (Krippendorff's α: *Constructiveness* 0.65, *Relevance* 0.68, *Specificity* 0.72, *Conciseness* 0.72), with all confidence intervals lying within $[0.60, 0.77]$ and significantly above chance. The human rankings (Table 2) mirror the automated ones: our approach outperforms every baseline on every metric, and the baseline ordering is preserved. Human ratings align strongly with `Prometheus` (Spearman $\rho = 0.85$), supporting its use as the automated evaluator throughout this paper.[13]

5.2 Ablation Study for Feedback Generation

We perform a series of ablation studies to establish the contribution of each aspect within our approach. *i. Effect of components within our approach*: We ablate our algorithm in Table 3 for Phi and find that removing the Template Construction phase drops Constructiveness by ~ 0.7 points for Phi, as templates provide structured scaffolds that reduce *review extension*. Removing the fitness function lowers scores by ~ 0.6 points and final candidate is often longer than its parents, revealing *length bias*, a common reward hacking shortcoming in LLM outputs (Emberson et al., 2025). We obtain similar results for the other models in this work.[14] *ii. Effect of individual rewards*: We analyze each reward in our *fitness function* (cf. Table 4) and find that removing the template adherence (sc_{temp}) reward drops Constructiveness by 0.6 points for Phi as outputs go off-topic. Removing the length-based reward (sc_{len}) reduces Conciseness by 0.6 points, reflecting how longer responses become convoluted and harder to read. Removing the cross-over component also leads to a decline in the constructiveness scores by 0.4 points pointing to the selection of less effective candidates. [15] *iii. Effect of planner inputs.* We ablate the three knowledge sources passed to the planner (paper abstract, reviewer-written summary, and reviewer-written strengths) in Table 5. Removing any single source reduces Phi's *Constructiveness* by 0.1–0.3 points, and removing two sources reduces it by up to 0.6 points, indicating that each input contributes complementary guidance rather than redundant signal. The effect is most pronounced for frequent, context-dependent issues such as "extra experiments," where grounding feed-

[13] Full alignment analysis in §A.6.1.

[14] Full ablation for all models in Table 29.

[15] Full ablation table in Tab 31

back requires the abstract for scope, the summary for the reviewer's framing, and the strengths to avoid contradicting acknowledged contributions.

6 Expert Assessment of Review Quality Improvement Through Feedback

Setup. To evaluate the quality of the generated feedback, we conduct a controlled study comparing the original reviews, rewrites based only on the detected *lazy thinking* and *specificity issues*, and rewrites that additionally incorporate the generated feedback. Following Purkayastha et al. (2025), we form two groups of two Ph.D. students each. One rewrites 50 reviews using only issues from our *issue detection* module emulating the setup in Purkayastha et al. (2025), while the other uses both issues and targeted feedback from our *feedback generation* approach, marking feedback as *actionable* or not. Both groups have access to the papers corresponding to the reviews. Rewrites are evaluated on *Constructiveness*, *Justified*, and *Adherence*. Two senior students perform pairwise comparisons between original reviews, issue-only rewrites, and issue-plus-feedback rewrites, splitting 50 reviews evenly and reserving 10 for agreement.

Results. We report the win-tie-loss results for the controlled experiment in Table 6. Reviews rewritten with feedback (Issue det. *w/* feed) outperform both reviews written with only *issue detection* (Issue det.) and the original reviews (Orig), achieving 85% and 95% higher Constructiveness, respectively. Even issue-only feedback improves over original reviews, consistent with Purkayastha et al. (2025). Reviewers in the feedback-based rewrite group incorporated 96% of the feedback, marking it actionable. Feedback-based rewrites reduce identified issues in reviews up to **92.4**%, compared to 85.2% for rewrites with only *issue detection*, demonstrating the effectiveness of the *feedback generation* module in driving review quality. Following Purkayastha et al. (2025), a Bradley-Terry preference model (Bradley and Terry, 1952) trained on adherence preferences yields scores of 1.2 for feedback-based rewrites, -1.0 for original reviews, and 0.6 for *issue detection*-based rewrites. This corresponds to a 93.4% win rate over original texts and 78.5% over issue-detection rewrites, further supporting the effectiveness of the annotations. Inter-annotator agreement (Krippendorff's α) is 0.71, 0.73, and 0.77 for Constructiveness, Justification, and Adherence, respectively.

Type	**Cons.** (W/T/L)	**Just.** (W/T/L)	**Adh.** (W/T/L)	**Δ Issue** (↑)
Issue det. vs. Orig	85/5/10	80/5/15	85/5/10	85.2
Issue det. *w* feed vs Orig	95/5/0	95/5/0	90/5/5	**92.4**
Issue det. *w* feed vs. Issu det.	85/10/5	90/5/5	85/10/5	78.5

Table 6: Comparison of Original, Issue-detected, and Issue-detected with feedback re-written reviews on Constructiveness, Justification, and Adherence using Win/Tie/Loss rates. **ΔIssue** indicates issue reduction.

7 Related Work

Review Quality Assessment. As scientific publications rise, reviewers face increasing burden, driving interest in assessing review quality (Kuznetsov et al., 2024). Prior studies have measured tonality (Bharti et al., 2024) and thoroughness (Lu et al., 2025). Sadallah et al. (2025) evaluate reviews based on constructiveness for author revisions, while Du et al. (2024) examine deficiencies in both human and LLM-generated reviews. In contrast, we focus on detecting cognitive biases (*lazy thinking*) and *specificity* issues following ACL ARR guidelines (Rogers and Augenstein, 2024), and propose a targeted feedback mechanism to improve review quality prior to submission.

LLM-generated feedback. LLMs have been widely used for feedback, especially in education, supporting English learners (Kasneci et al., 2023; Escalante et al., 2023), coding instruction (Misiejuk et al., 2024; Azaiz et al., 2024), and TOEFL preparation (Kasneci et al., 2023). Studies show LLM feedback matches human quality and drives positive outcomes. Building on this, Thakkar et al. (2025) used Claude at ICLR 2025 to improve review specificity, correctness, and politeness. However, their method ignores guideline compliance and relies on a closed-source model. We address both by enforcing ACL ARR adherence and using open-weight LLMs, ensuring integrity and privacy.

8 Conclusion

To enhance review quality, we introduce an open-weight, LLM-driven framework that identifies issues precisely and generates targeted feedback via iteratively refined templates. It outperforms zero-shot and fine-tuned LLM-based baselines, and explicit feedback further improves review quality. Additionally, we release LAZYREVIEWPLUS, an expert-annotated dataset of review sentences labeled for *lazy thinking* and *specificity*. We hope our work inspires further efforts to enhance the quality of peer review.

Limitations

In this work, we propose a new dataset alongside a precision-oriented issue identification and feedback generation framework designed to enhance the quality of peer reviews. However, our approach comes with several limitations.

First, we introduce LAZYREVIEWPLUS, a single-segment, multi-label classification dataset for detecting *lazy thinking* and *specificity* issues in peer reviews. The dataset currently covers only conferences hosted on ACL Rolling Review (ARR), due to their well-defined guidelines for *lazy thinking* and *specificity*. Extending this resource to other domains, conferences, or languages would require developing equally rigorous guidelines with precise definitions for these issues.

Second, in terms of methodology, we conduct extensive experiments and analyses for both **issue identification** and **feedback generation**. However, these results may not be fully generalizable to other scientific or evaluative tasks, and thus the results should be interpreted with caution. Moreover, our feedback generation approach relies on predefined templates that are refined using a genetic algorithm. Over time, however, reviewers may become desensitized to recurring feedback styles. The long-term effectiveness and robustness of a template-driven approach therefore warrant further investigation, especially in repeated or large-scale deployment settings. Additionally, deploying our approach requires curating templates for each conference upon the introduction of new guidelines. Nevertheless, curating 25 templates for the ACL Rolling Review guidelines required only 250 minutes — a modest one-time overhead relative to the substantial resources involved in organizing a conference. Moreover, while addressing *lazy thinking* and improving *specificity* is an attempt to enhance review quality, overall scientific outcomes depend on multiple confounding factors, including paper quality, author rebuttals, and meta-reviewer decisions. Evaluating the long-term impact of guideline-compliant reviews on acceptance decisions and scientific progress remains an important direction for future work.

Third, our framework relies on large language models, which may carry stylistic or cultural biases inherited from their training data. Although peer reviewing is intended to be domain-neutral, such biases may still manifest subtly in phrasing or prioritization. Future work could incorporate bias detection and mitigation strategies to enhance fairness and neutrality in feedback generation.

Finally, while automated feedback systems can assist reviewers and improve the clarity and constructiveness of reviews, they should complement—rather than replace—human judgment. The ultimate responsibility for fair, constructive, and contextually informed reviewing must remain with human reviewers.

Ethics Statement

Before commencing the project, we obtained the necessary ethics approvals from the Ethics Committees of the leading institute. Our dataset, LAZYREVIEWPLUS, will be released under the CC-BY-NC 4.0 license. The underlying ARR reviews were collected from NLPEER (Dycke et al., 2023), which is licensed under the same terms. The analysis and automatic annotation processes do not involve any personal or sensitive information. The annotators participating in the user study and dataset annotation were compensated at a rate of $25 per hour, ensuring fair remuneration.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, et al. 2024. Phi-4 technical report. *ArXiv preprint arXiv: 2412.08905*.

ACL Publication Ethics Committee. 2024. ACL Policy on Publication Ethics. *Association for Computational Linguistics*.

Balazs Aczel, Barnabas Szaszi, and Alex O Holcombe. 2021. A billion-dollar donation: estimating the cost of researchers' time spent on peer review. *Research integrity and peer review*, 6(1):1–8. https://link.springer.com/article/10.1186/s41073-021-00118-2.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, et al. 2025. gpt-oss-120b & gpt-oss-20b model card. *Arxiv preprint arXiv: 2508.10925*.

Imen Azaiz, Natalie Kiesler, and Sven Strickroth. 2024. Feedback-Generation for Programming Exercises with GPT-4. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, ITiCSE 2024, page 31–37, New York, NY, USA. Association for Computing Machinery.

Prabhat Kumar Bharti, Meith Navlakha, Mayank Agarwal, and Asif Ekbal. 2024. PolitePEER: does peer review hurt? A dataset to gauge politeness intensity in the peer reviews. *Language Resources and Evaluation*, 58(4):1291–1313. https://doi.org/10.1007/s10579-023-09662-3.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, et al. 2024. DeepSeek LLM: Scaling Open-Source Language Models with Longtermism. *Arxiv preprint arXiv: 2401.02954*.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345. https://doi.org/10.1093/biomet/39.3-4.324.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *Arxiv preprint arXiv: 2407.21787*.

Jiangshu Du, Yibo Wang, Wenting Zhao, Zhongfen Deng, Shuaiqi Liu, Renze Lou, Henry Peng Zou, Pranav Narayanan Venkit, Nan Zhang, Mukund Srinath, Haoran Ranran Zhang, Vipul Gupta, Yinghui Li, Tao Li, Fei Wang, Qin Liu, Tianlin Liu, Pengzhi Gao, Congying Xia, Chen Xing, Cheng Jiayang, Zhaowei Wang, Ying Su, Raj Sanjay Shah, Ruohao Guo, Jing Gu, Haoran Li, Kangda Wei, Zihao Wang, Lu Cheng, Surangika Ranathunga, Meng Fang, Jie Fu, Fei Liu, Ruihong Huang, Eduardo Blanco, Yixin Cao, Rui Zhang, Philip S. Yu, and Wenpeng Yin. 2024. LLMs assist NLP researchers: Critique paper (meta-)reviewing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5081–5099, Miami, Florida, USA. Association for Computational Linguistics.

Nils Dycke, Ilia Kuznetsov, and Iryna Gurevych. 2022. Yes-yes-yes: Proactive data collection for ACL rolling review and beyond. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 300–318, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Luke Emberson, Ben Cottier, Josh You, Tom Adamczewski, and Jean-Stanislas Denain. 2025. LLM responses to benchmark questions are getting longer over time. *Blog Post*.

Juan Escalante, Austin Pack, and Alex Barrett. 2023. AI-generated feedback on writing: Insights into efficacy and ENL student preference. *International Journal of Educational Technology in Higher Education*, 20(1):57. https://doi.org/10.1186/s41239-023-00425-2.

Tao Feng, Lizhen Qu, and Gholamreza Haffari. 2023. Less is more: Mitigate spurious correlations for open-domain dialogue response generation models by causal discovery. *Transactions of the Association for Computational Linguistics*, 11:511–530.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221–233. https://doi.org/10.1037/h0057532.

David E Goldberg. 1990. A note on boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing. *Complex Systems*, 4:445–460.

Junjie Hu, Yu Cheng, Zhe Gan, Jingjing Liu, Jianfeng Gao, and Graham Neubig. 2020. What Makes A Good Story? Designing Composite Rewards for Visual Storytelling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7969–7976.

Fantine Huot, Joshua Maynez, Shashi Narayan, Reinald Kim Amplayo, Kuzman Ganchev, Annie Priyadarshini Louis, Anders Sandholm, Dipanjan Das, and Mirella Lapata. 2023. Text-blueprint: An interactive platform for plan-based conditional generation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 105–116, Dubrovnik, Croatia. Association for Computational Linguistics.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, et al. 2024. GPT-4o System Card. *Arxiv preprint arXiv: 2410.21276*.

Thorben Jansen, Lars Höft, J Luca Bahr, Livia Kuklick, and Jennifer Meyer. 2025. Constructive feedback can function as a reward: Students' emotional profiles in reaction to feedback perception mediate associations with task interest. *Learning and Instruction*, 95:102030. https://doi.org/10.1016/j.learninstruc.2024.102030.

Daniel P Jeong, Zachary Chase Lipton, and Pradeep Kumar Ravikumar. 2025. LLM-select: Feature selection with large language models. *Transactions on Machine Learning Research*, 2025.

Xiaoqiang Kang, Zimu Wang, Xiaobo Jin, Wei Wang, Kaizhu Huang, and Qiufeng Wang. 2025. Template-Driven LLM-Paraphrased Framework for Tabular Math Word Problem Generation. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence, February 29-March 4, 2025, Philadelphia, Pennsylvania, USA*, pages 24303–24311. AAAI Press.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274. https://doi.org/10.1016/j.lindif.2023.102274.

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

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

Klaus Krippendorff. 2004. Reliability in content analysis: Some common misconceptions and recommendations. *Human communication research*, 30(3):411–433. https://doi.org/10.1111/j.1468-2958.2004.tb00738.x.

Ilia Kuznetsov, Osama Mohammed Afzal, Koen Dercksen, Nils Dycke, Alexander Goldberg, Tom Hope, Dirk Hovy, Jonathan K. Kummerfeld, Anne Lauscher, et al. 2024. What Can Natural Language Processing Do for Peer Review? *Arxiv preprint arXiv: 2405.06563*.

Mengfei Lan, Lecheng Zheng, Shufan Ming, and Halil Kilicoglu. 2024. Multi-label sequential sentence classification via large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16086–16104, Miami, Florida, USA. Association for Computational Linguistics.

Kuang-Huei Lee, Ian Fischer, Yueh-Hua Wu, Dave Marwood, Shumeet Baluja, Dale Schuurmans, and Xinyun Chen. 2025. Evolving Deeper LLM Thinking. *Arxiv Preprint arXiv: 2501.09891*.

Weixin Liang, Yaohui Zhang, Zhengxuan Wu, Haley Lepp, Wenlong Ji, Xuandong Zhao, Hancheng Cao, Sheng Liu, Siyu He, Zhi Huang, Diyi Yang, Christopher Potts, Christopher D Manning, and James Y. Zou. 2024. Mapping the increasing use of LLMs in scientific papers. In *First Conference on Language Modeling, University of Pennsylvania, Philadelphia, PA, October 7-9, 2024*. Openreview.net.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *Arxiv preprint arXiv: 1907.11692*.

Sheng Lu, Ilia Kuznetsov, and Iryna Gurevych. 2025. Identifying aspects in peer reviews. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6145–6167, Suzhou, China. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, et al. 2023. Self-Refine: Iterative Refinement with Self-Feedback. In *Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems, NeurIPS 2023, New Orleans, Louisiana, USA, Dec 10-16, 2023*. Openreview.net.

Hariharan Manikandan, Yiding Jiang, and J Zico Kolter. 2023. Language models are weak learners. In *Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems, NeurIPS 2023, New Orleans, Louisiana, USA, Dec 10-16, 2023*. Openreview.net.

Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025. Unifying AI tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of LLM-powered AI tutors. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1234–1251, Albuquerque, New Mexico. Association for Computational Linguistics.

Kamila Misiejuk, Rogers Kaliisa, and Jennifer Scianna. 2024. Augmenting assessment with AI coding of online student discourse: A question of reliability. *Computers and Education: Artificial Intelligence*, 6:100216. https://doi.org/10.1016/j.caeai.2024.100216.

Shashi Narayan, Joshua Maynez, Reinald Kim Amplayo, Kuzman Ganchev, Annie Louis, Fantine Huot, Anders Sandholm, Dipanjan Das, and Mirella Lapata. 2023. Conditional generation with a question-answering blueprint. *Transactions of the Association for Computational Linguistics*, 11:974–996.

Viet Thanh Pham, Lizhen Qu, Zhuang Li, Suraj Sharma, and Gholamreza Haffari. 2025. SurveyPilot: an agentic framework for automated human opinion collection from social media. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4397–4422, Vienna, Austria. Association for Computational Linguistics.

Axel Pichler, Janis Pagel, and Nils Reiter. 2025. Evaluating LLM-prompting for sequence labeling tasks in computational literary studies. In *Proceedings of the 9th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2025)*, pages

32–46, Albuquerque, New Mexico. Association for Computational Linguistics.

Sukannya Purkayastha, Anne Lauscher, and Iryna Gurevych. 2023. Exploring jiu-jitsu argumentation for writing peer review rebuttals. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14479–14495, Singapore. Association for Computational Linguistics.

Sukannya Purkayastha, Zhuang Li, Anne Lauscher, Lizhen Qu, and Iryna Gurevych. 2025. LazyReview: A dataset for uncovering lazy thinking in NLP peer reviews. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3280–3308, Vienna, Austria. Association for Computational Linguistics.

L. A. Ramshaw and M. P. Marcus. 1999. *Text Chunking Using Transformation-Based Learning*, pages 157–176. Springer Netherlands, Dordrecht. https://doi.org/10.1007/978-94-017-2390-9_10.

Anna Rogers, Marzena Karpinska, Jordan Boyd-Graber, and Naoaki Okazaki. 2023. Program chairs' report on peer review at acl 2023. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages xl–lxxv, Toronto, Canada. Association for Computational Linguistics.

Anne Rogers and Isabelle Augenstein. 2024. How to review for ACL Rolling Review. *ACL Rolling Review*.

Abdelrahman Sadallah, Tim Baumgärtner, Iryna Gurevych, and Ted Briscoe. 2025. The good, the bad and the constructive: Automatically measuring peer review's utility for authors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28979–29009, Suzhou, China. Association for Computational Linguistics.

Dhruv Sahnan, David Corney, Irene Larraz, Giovanni Zagni, Ruben Miguez, Zhuohan Xie, Iryna Gurevych, Elizabeth Churchill, Tanmoy Chakraborty, and Preslav Nakov. 2025. Can LLMs Automate Fact-Checking Article Writing? *Arxiv preprint arXiv: 2503.17684 (TACL, to appear)*.

Mohammad Mustafa Taye. 2023. Understanding of machine learning with deep learning: architectures, workflow, applications and future directions. *Computers*, 12(5):91.

Nitya Thakkar, Mert Yuksekgonul, Jake Silberg, Animesh Garg, Nanyun Peng, Fei Sha, Rose Yu, Carl Vondrick, and James Zou. 2025. Can LLM feedback enhance review quality? A randomized study of 20K reviews at ICLR 2025. *Arxiv preprint arXiv: 2504.09737*.

Hendrik P. van Dalen and Kène Henkens. 2012. Intended and unintended consequences of a publish-or-perish culture: A worldwide survey. *Journal of the American Society for Information Science and Technology*, 63(7):1282–1293. https://doi.org/10.1002/asi.22636.

Mark Ware and Michael Mabe. 2015. An overview of scientific and scholarly journal publishing. *The STM Report*, page 1082:1083. https://digitalcommons.unl.edu/scholcom/9/.

Qiyao Wei, Samuel Holt, Jing Yang, Markus Wulfmeier, and Mihaela van der Schaar. 2025. The AI Imperative: Scaling High-Quality Peer Review in Machine Learning. *Arxiv preprint arXiv: 2506.08134*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xiaotong (Tone) Xu, Jiayu Yin, Catherine Gu, Jenny Mar, Sydney Zhang, Jane L. E, and Steven P. Dow. 2024. Jamplate: Exploring LLM-Enhanced Templates for Idea Reflection. In *Proceedings of the 29th International Conference on Intelligent User Interfaces, IUI '24, March 18-21, 2024, Greenville, SC, USA*, page 907–921. Association for Computing Machinery.

Antoun Yaacoub, Zainab Assaghir, Lionel Prevost, and Jérôme Da-Rugna. 2025. Analyzing Feedback Mechanisms in AI-Generated MCQs: Insights into Readability, Lexical Properties, and Levels of Challenge. *Arxiv preprint arXiv: 2504.21013*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2025. Qwen2.5 technical report. *Arxiv preprint arXiv: 2412.15115*.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, et al. 2025. Yi: Open foundation models by 01.ai. *ArXiv preprint arXiv: 2403.04652*.

A Appendix

A.1 Choice of LLMs and implementation details

Following Purkayastha et al. (2025), we select models that are privacy-preserving in accordance with ARR publication ethics guidelines (ACL Publication Ethics Committee, 2024) and of manageable sizes for practical deployment. All the LLMs are implemented using the vLLM library and huggingface transformers (Wolf et al., 2020).[16]. For calculating readability, we use the `textstat` package.[17] The experiments are run on a single A100 GPU and none of the experiments consumed more than 36 hours.

A.2 Annotation Guidelines

A.2.1 Task Description

As the volume of scientific publications grows, maintaining high-quality peer review becomes increasingly important. One common issue in reviews is the presence of non-specific reviews, which makes it difficult for authors and program chairs to address concerns effectively. Your task is to identify such non-specific issues in paper reviews and classify them into predefined categories.

Due to the high reviewing load, the reviewers are often prone to different kinds of bias, which inherently contribute to lower reviewing quality and hampers scientific progress in the field. The ACL peer reviewing guidelines characterize some of such reviewing bias in the table below, known as lazy thinking. The table mentions the heuristic and the reason for the statement being problematic. In this task, we aim to annotate these biased sentences in peer-reviews from scientific conferences.

A.2.2 Typical *Lazy Thinking* and *Specificity Issues*

Some specificity and lazy thinking issues are shown in Table 7. The full *specificity* issues table is in Table 10 and *lazy thinking* in Table 11 respectively.

A.2.3 Outlining differences within Summary of Weaknesses and Comments, Suggestions, and Typos

Summary of Weaknesses: This field is often naively interpreted as the reasons to reject. ARR review guidelines provide the following definition:

[16] https://docs.vllm.ai/en/latest/

[17] https://pypi.org/project/textstat/

	Issue	Meaning / Phrasing
Lazy Thinking	Results are not novel	If the paper claims e.g., a novel method, and you think you've seen this before, you need to provide a reference.
	The topic is too niche	A main track paper may well make a big contribution to a narrow subfield.
Specificity	The contribution is not novel	Highly similar work X and Y has been published 3+ months prior to the submission deadline.
	X is not clear	Y and Z are missing from the description of X.

Table 7: Examples of peer review issues categorized as *lazy thinking* and *specificity* as per ACL ARR guidelines (Rogers and Augenstein, 2024).

> What are the concerns that you have about the paper that would cause you to favor prioritizing other high-quality papers that are also under consideration for publication? These could include concerns about correctness of the results or argumentation, limited perceived impact of the methods or findings (note that impact can be significant both in broad or in narrow sub-fields), lack of clarity in exposition, or any other reason why interested readers of *ACL* papers may gain less from this paper than they would from other papers under consideration. Where possible, please number your concerns so authors may respond to them individually.

This field should contain reasons that make the paper not ready for publication at the current stage.

Comments, Suggestions, and Typos: These are additional suggestions that reviewers can provide to authors to improve the manuscript. ARR review guidelines define this field as:

> If you have any comments to the authors about how they may improve their paper, other than addressing the concerns above (weakness), please list them here.

The major difference between these two fields is the **severity of the issue**.

A.2.4 Annotation Instructions for Review Sentences

Method	Model	Prec.	Rec.	F1
Sequential	Yi	0.72	0.71	0.72
	Phi	0.73	0.72	0.72
	Qwen	0.67	0.61	0.64
	Deep.	0.62	0.62	0.62
	Oss.	0.71	0.68	0.69
Standalone	Yi	0.78	0.76	0.77
	Phi	**0.81**	**0.77**	**0.79**
	Qwen	0.73	0.67	0.70
	Deep.	0.72	0.66	0.69
	Oss.	0.75	0.73	0.73

Table 8: Performance comparison of various models across different methods on the segment detection task, evaluated in terms of Precision (Prec.), Recall (Rec.), and F1.

To identify review segments, we first determine the boundary of each sentence using BIO tags. Given a review sentence, we classify it as the Beginning (B), Inside (I) of a review segment, or Other (O). We consider the preceding sentence: if it relates to a different topic, the current sentence is marked as B; if it continues the previous sentence, it is marked as I; and if it is not relevant (e.g., punctuation or non-substantial content), it is marked as O.

Next, we identify the section of the review sentence. Each sentence is categorized as belonging to the Summary of Weaknesses, Comments and Suggestions, or Summary of Strengths. According to ARR guidelines, major criticisms that influence acceptance decisions are placed in the Summary of Weaknesses, while other suggestions or minor criticisms belong in Comments and Suggestions.

Finally, we identify lazy thinking or non-specific writing using multi-label classification. For each sentence, we assign appropriate categories of lazy thinking (e.g., "the results are not novel") and non-specificity (e.g., "X was done in the way Y"). The entire segment, formed by all sentences tagged with B/I together, is considered before assigning labels. For example, the sentences "I do not understand the setup" (B) and "There is no mention of epochs or hyperparameters in the dataset" (I) together form a segment:

> "I do not understand the setup. There is no mention of epochs or hyperparameters in the dataset."

The segment collectively contributes to the label, e.g., **None**.

A.3 Analysis of the dataset, LAZYREVIEWPLUS

We show instances from our dataset in Table 9. We show the distribution of sentences per segment in Fig 4a.The distribution of labels within each segment in Fig 4b. The distribution of *specificity* and *lazy thinking* issues in Figures 2a and **??** respectively.

A.4 Segment Identification

Approach We perform segment detection using two approaches: **1. Standalone**: For each sentence s_i in review R, predict $\hat{y}i = f\theta(s_i, R)$ to indicate beginning (B), inside (I), or outside (O) a segment. This relies solely on s_i contextualized within the full review. **2. Sequential**: To capture inter-sentence dependencies, the LLM conditions on prior predictions $\hat{Y} < i = (\hat{y}1, \ldots, \hat{y}i - 1)$, predicting $\hat{y}i = f\theta(s_i, R, \hat{Y} < i)$. This provides temporary memory of past tagging decisions.

Prompts. The prompt used for standalone and sequential segment identification is provided in Fig 5 and Fig 6 respectively.

Overall Results. We present the performance of all models across different approaches in Table 8. The **standalone** strategy, which contextualizes each review segment within the full review, emerges as the most effective. In contrast, the **sequential** strategy suffers from error propagation, as reliance on prior predictions leads to cascading misclassifications, and sporadically may also introduces spurious correlations, consistent with prior findings (Feng et al., 2023; Purkayastha et al., 2025). Among the models, Phi consistently outperforms the others, which can be attributed to its specialized training on textbook-quality corpora combined with a balanced mix of synthetic and real-world data (Abdin et al., 2024).

Error Analysis. We analyze the performance of all models using the best-performing approach, standalone. We observe that the top model, Phi, occasionally confuses B tags with I tags. This pattern of confusion is consistent across the other models in our study (cf. Fig 8). This behavior is intuitive, as some review comments can be interpreted as standalone statements instead of occurring within the broader context. Nevertheless, the presence of extra 'B' tags does not substantially hinder **issue detection**, since our segment-level annotations

Review Segment	*Lazy Thinking*	*Specificity*
More experiments on datasets are needed, and I would also like to see some other LLMs, like the Gemini family, also be evaluated to make the paper more comprehensive and generalizable.	Authors could also do extra experiment; Authors should compare to a closed model	N/A
Although the paper is interesting and strongly supported by experimental results, understanding the writing is tough.	The paper has language errors	X is not clear
Results in Table 1 are lower than SOTA results, casting doubt on the usefulness and novelty of this approach.	Results do not surpass the latest SOTA; Results are not novel	The contribution is not novel

Table 9: Instances from our dataset, LAZYREVIEWPLUS where each segment may have multiple *lazy thinking* and *specificity* issues as per the ACL ARR guidelines (Rogers and Augenstein, 2024)

Non-Specific Issues	Specific Rewrite
The paper is missing relevant references	The paper is missing references XYZ
X is not clear	Y and Z are missing from the description of X.
The formulation of X is wrong	The formulation of X misses the factor Y
The contribution is not novel	Highly similar work X and Y has been published 3+ months prior to submission deadline
The paper is missing recent baselines	The proposed method should be compared against recent methods X, Y and Z (see H14 below for requesting comparisons to 'closed' systems)
X was done in the way Y	X was done in the way Y which has the disadvantage Z
The algorithm's interaction with dataset is problematic	It's possible that when using the decoding (line 723) on the dataset 3 (line 512), there might not be enough training data to rely on the n-best list.[If reasonably well-known entities are discussed]

Table 10: Examples of *specificity* issues and their corresponding specific rewrites.

ensure that each segment still corresponds to annotator detected issues within the dataset.

A.5 Issue Identification

A.5.1 Zero-shot Issue Identification

The zero-shot results from the best-performing LLM, GPT-OSS, fail to capture several classes that are sufficiently frequent in our dataset (cf. Fig. 10). While this type of classification achieves high precision for some classes, it introduces substantially more errors than a controlled approach.

A.5.2 Baselines

For **issue detection**, we employ the following baselines: (i) Zero-shot ($\texttt{LLM}_{\texttt{Zero}}$): the LLM predicts $l_i \subseteq \mathcal{L}$ given s_i; (ii) Finetuning ($\texttt{LLM}_{\texttt{Fine}}$): we finetune the LLM on the training split and evaluate with cross-validation; (iii) QA-based classification ($\texttt{LLM}_{\texttt{QA}}$ / $\texttt{LLM}_{\texttt{QAFine}}$): the pretrained LLM ($\texttt{LLM}_{\texttt{QA}}$) or its finetuned variant ($\texttt{LLM}_{\texttt{QAFine}}$) classifies s_i using the same feature-specific questions and answers as in our approach. (iv) Encoder-Only Baseline: We use RoBERTa (Liu et al., 2019) as a baseline, trained on the same data as the $\texttt{LLM}_{\texttt{Fine}}$ model, to evaluate the impact of task-specific finetuning using smaller models.

A.5.3 Prompts for Issue Identification

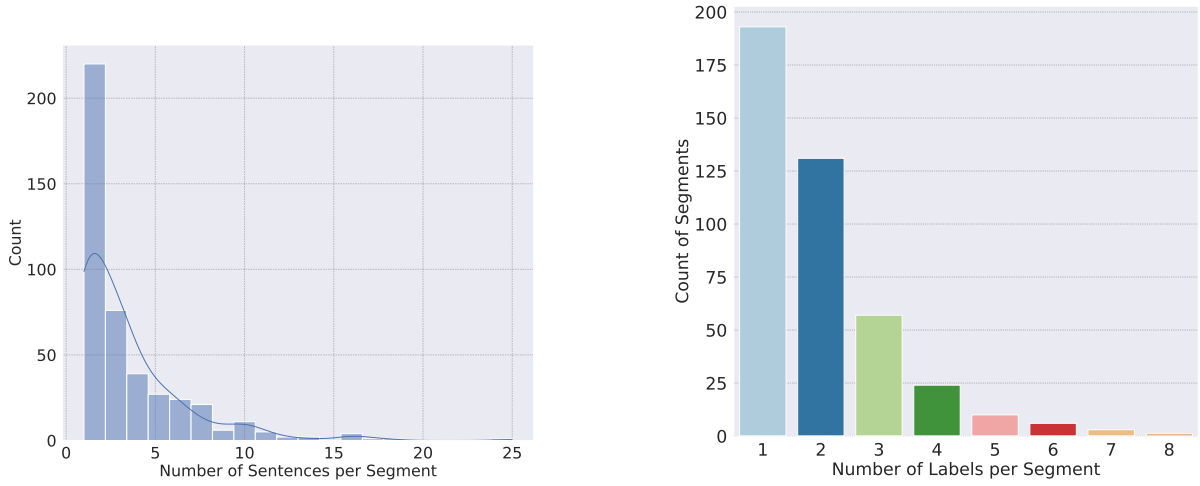
We present the **issue identification** prompt in Fig 9. The feature extraction prompt extracts abstract-level feature and converts it into the feature question for the LLM to answer using the Feature QA prompt. The response from the Feature QA prompt is mapped to the discrete values: Yes $\rightarrow$ 1, No $\rightarrow$ -1, and Other $\rightarrow$ 0. We show a running example of a segment and abstract feature questions in Fig 11.

Heuristic	Why this is problematic
H1. The results are not surprising	Many findings seem obvious in retrospect, but this does not mean that the community is already aware of them and can use them as building blocks for future work. Some findings may seem intuitive but haven't previously been tested empirically.
H2. The results contradict what I would expect	You may be a victim of confirmation bias, and be unwilling to accept data contradicting your prior beliefs.
H3. The results are not novel	If the paper claims e.g. a novel method, and you think you've seen this before - you need to provide a reference (note the policy on what counts as concurrent work). If you don't think that the paper is novel due to its contribution type (e.g. reproduction, reimplementation, analysis) — please note that they are in scope of the CFP and deserve a fair hearing.
H4. This has no precedent in the existing literature	Papers that are more novel tend to be harder to publish. Reviewers may be unnecessarily conservative.
H5. The results do not surpass the latest SOTA	SOTA results are neither necessary nor sufficient for a scientific contribution. An engineering paper could also offer improvements on other dimensions (efficiency, generalizability, interpretability, fairness, etc.) If the authors do not claim that their contribution achieves SOTA status, the lack thereof is not an issue.
H6. The results are negative	The bias towards publishing only positive results is a known problem in many fields, and contributes to hype and overclaiming. If something systematically does not work where it could be expected, the community does need to know about it.
H7. This method is too simple	The goal is to solve the problem, not to solve it in a complex way. Simpler solutions are preferable, as they are less brittle and easier to deploy in real-world settings.
H8. The paper doesn't use [my preferred methodology]	NLP is an interdisciplinary field, relying on many kinds of contributions: models, resource, survey, data/linguistic/social analysis, position, and theory.
H9. The topic is too niche	A main track paper may well make a big contribution to a narrow subfield.
H10. The approach is tested only on [not English]	The same is true of NLP research that tests only on English. Monolingual work on any language is important practically (methods/resources) and theoretically (deeper understanding of language).
H11. The paper has language errors	As long as the writing is clear enough, better scientific content should be more valuable than better journalistic skills.
H12. The paper is missing the [reference X]	Per ACL policy, missing references to prior highly relevant work is a problem if such work was published 3+ months before the submission deadline. Otherwise, missing references belong in the "suggestions" section.
H13. The authors could also do [extra experiment X]	It is always possible to come up with extra experiments and follow-up work. But a paper only needs to present sufficient evidence for the claim that the authors are making. Extra experiments belong in the "nice-to-have" category rather than "reasons to reject."
H14. The authors should compare to a 'closed' model X	Requesting comparisons to closed-source models is only reasonable if it directly bears on the claim the authors are making. Many important questions require more openness than closed models allow.
H15. The authors should have done [X] instead	"I would have written this paper differently." This criticism applies only if the authors' choices prevent them from answering their research question or their framing is misleading. Otherwise, it is just a suggestion, not a weakness.
H16. Limitations != weaknesses	No paper is perfect, and most CL venues now require a Limitations section. A good review should not list limitations as reasons to reject unless appropriately motivated.

Table 11: *Lazy Thinking* issues as per ARR Reviewer guidelines (Rogers and Augenstein, 2024).

A.5.4 Data Split for Issue Identification

We show the distributions of labels within train and test by employing various methods in Fig 7.



(a) Sentences within a segment

(b) Distribution of labels within segments

Figure 4: Overview of sentence and label distributions in our proposed dataset, LAZYREVIEWPLUS.

Standalone Segment Detection Prompt

Instruction:
You are a Segment Tagger. Your task is to tag the sentence with one of these labels in the context of the review. A segment consists of multiple sentences. The first sentence should be tagged with "B" and the remaining sentences with "I". Carefully examine where the sentence appears in the review before assigning a label.

Labels:
B: Beginning of a segment.
I: Continuation of a segment. Always follows a B.

Review: `{review}`

Sentence: `{sentence}`

Output:
Answer with *only one label*: B or I.

Figure 5: Prompt used for standalone segment detection.

Random review-level splits result in poor label balance, whereas our approach achieves nearly the same label balance as the `sklearn` train-test split. However, the `sklearn` split does not account for real-world scenarios, where full reviews are used as input for issue identification. In contrast, our distribution-aware split maintains label balance for effective cross-validation while also reflecting realistic use cases. We also show the distance across various split methods in Table 14.

A.5.5 Results

Overall results. We report overall performances in Table 13. Our neurosymbolic approach, combining LLM-extracted features with a classical ML classifier, achieves the best results across the board, surpassing zero-shot performance by at least **0.9** points. Notably, `RoBERTa` outperforms all LLM-based baselines, reaffirming that task-specific fine-tuning with smaller models can be more effective than LLMs with limited data. Fine-tuning LLMs on our dataset yields meaningful gains (0.04–0.23 in Yi), highlighting the value of a specialized dataset, yet still falls short of our approach due to LLMs' high data requirements for task adaptation. $\texttt{LLM}_{\texttt{QA}}$ improves over zero-shot variants, confirming the value of feature-aligned questioning, while $\texttt{LLM}_{\texttt{QAFine}}$ underperforms, likely due to data-constrained segment-level finetuning. Overall, Extra Trees with Phi features delivers the strongest **issue detection**, benefiting from structured neurosymbolic representations and tree-based non-linear modeling. Low variance (below 0.01) further confirms dataset robustness (cf. Table 22). Additionally, smaller models such as DeepSeek and Phi achieve better performance than larger LLMs which demonstrates that our framework is not criti-

Sequential Segment Detection Prompt

Instruction:
You are a Segment Tagger. Your task is to tag the sentence with one of these labels in the context of the review. A segment consists of multiple sentences. The first sentence should be tagged with "B" and the remaining sentences with "I". Carefully examine where the sentence appears in the review before assigning a label.

Labels:
B: Beginning of a segment.
I: Continuation of a segment. Always follows a B.
O: Outside any segment.

Review: `{review}`

Previous sentence tag: `{previous_prediction}`

Sentence: `{sentence}`

Output:
Answer with *only one label*: B, I, or O.

Figure 6: Prompt used for sequential segment detection.

cally dependent on a single architecture or scale.

A.5.6 Ablation Study

We conduct a series of ablation experiments to evaluate the effectiveness of our approach: **1. Representation Effectiveness:** We employed a sentence embedding model to encode review segments into vector representations and evaluated their effectiveness for issue classification. Table 17 reports the performance of different classifiers on the issue detection task using embeddings from the `all-MiniLM-L6-v2` sentence transformer, while Table 19 compares these embedding-based approaches with our proposed feature-vector method under a 50% train–test split. Table 18 presents the performance of classifiers using feature vectors constructed from five questions on Phi. Precision and recall results for the baselines discussed in §4 are provided in Table 21. Overall, even the best-performing embedding-based classifier performs approximately **0.5** macro $F_{0.5}$ points worse than our proposed feature-vector-based approach. **2. Feature question size**: We evaluate the impact of reducing the feature vector size from 10 to 5 using our best-performing model, Phi, due to its superior feature extraction capabilities. We find that reducing the feature vector leads to a **0.12** point drop in F0.5 score in cross-validation[18]. This indicates that the smaller feature set fails to capture sufficient information, likely causing underfitting and limiting the model's representational capacity. **3. Effect of various thresholds**: We evaluate our approach against the strongest baseline, `RoBERTa`, across thresholds $\{0.25, 0.5, 0.75, 1.0, 2.0\}$ (Table 20). At $\beta = 0.25$, Extra Trees with Phi features achieves high precision (0.90) but low recall (0.24). Increasing β (0.5–2.0) boosts recall at some precision loss (e.g., 0.42/0.67 at $\beta = 2.0$), showing tunable precision–recall trade-offs. Fig. 12 shows our approach consistently outperforms this strongest baseline, providing reliable signals while balancing precision and recall. **4. Human-written questions**. We also experimented with manually crafted questions for each issue type. Designing 10 questions per issue required approximately 45 minutes, resulting in a total effort of over 20 hours for all 27 issue types. The corresponding results are reported in Table 23. Using human-written questions, Extra Trees with Phi achieves an F1-score of 0.57, compared to 0.51 obtained with LLM-generated questions (cf. Table 13). However, review guidelines frequently evolve, making manual question engineering costly and difficult to maintain. In contrast, our LLM-based question generation framework provides a scalable and economical alternative that can readily adapt to changing review standards. Therefore, the modest performance gain from human-written questions does not outweigh the practical advantages of automated LLM-driven question generation.

[18] Details in Table 18.

A.6 Additional details on feedback generation

A.6.1 Alignment Study with Prometheus and other evaluators

To assess the reliability of our reference-free automated metric, we conducted an alignment study comparing its scores with those provided by Prometheus and human annotators. The results, summarized in Table 24, show an average Spearman correlation of **0.85**, demonstrating the strong effectiveness of the metric. We further compare the performance of Prometheus v1, Prometheus v2 and GPT 4o in Table 25. Prometheus V2 outperforms all the other methods across the board in terms of Kendall τ.

A.6.2 Experimental Setup and Analysis for review rewrites and role extension

To identify occurrences of review extension and role separation in LLM-generated feedback, we use GPT-OSS 120B (Agarwal et al., 2025) as a judge, prompting it to detect such issues within zero-shot generations using the instructions in Table 26. The alignment between GPT-OSS judgments and human evaluations yields a Spearman correlation of 0.85. Table 26 presents the results, illustrating the prevalence of role separation' and review extension' in most of LLM-generated feedback. Example cases of role separation and review extension are shown in Fig. 14 and Fig. 13, respectively.

A.6.3 List of forbidden terms in the fitness function

To ensure that feedback remains professional, actionable, and relevant, we identify greetings and off-topic expressions that should be avoided in review comments. Such terms do not contribute to the evaluation of the work and may distract or confuse the author. In our zero-shot feedback generations, we found that approximately 82% of the divergent feedback contained one or more of these terms. Consequently, we compiled them into a list of forbidden terms to promote professionalism and maintain focus in reviewer feedback. `forbidden_terms = ["Hi", "Hello", "Hey", "Dear Author", "To whom it may concern", "Greetings", "Good morning", "Good afternoon", "Good evening", "Haha", "Hehe", "Lmao", "OMG", "Wow", "FYI", "Cheers", "Best regards", "Sincerely", "I think"]`

A.6.4 Prompts for feedback generation

Prompt for Prometheus The Conciseness prompt is in Fig 15, Relevance is in Fig 16, Specificity is in Fig 18 and Constructiveness is in Fig 17 respectively.

1-pass or zero-shot. The prompt is provided in Fig 19.

Plan Generation. The plan generation prompt is in Fig 21.

Template (Temp). The prompt is provided in Fig 20

Plan-then-generate (Plan) The prompt is provided in Fig 22.

Self Refine. (Self Ref.) The 1-pass prompt is first run to create the initial feedback. Then the self-refine prompt in Fig 23 is run n_{gen} times to get the final feedback.

Genetic Algorithm. The genetic algorithm first uses Plan-then-generate prompt to generate candidates in Fig 22 multiple times which are scored via the fitness function. The top $n_{parents}$ selected then go throug crossover to generate new offsprings as in Fig 24. The new candidate pool then goes through n_{gen} iterations to generate the final candidate.

A.6.5 Hyper-parameters used

Following prior work on using evolutionary algorithm for generating responses (Lee et al., 2025; Pham et al., 2025). we set the hyper-parameters as in Table 27.

A.6.6 Evaluation of the generated feedback

We present the full results of the automated evaluation using Prometheus in Table 16 and human evaluation (cf. §5) of the feedback in Table 2.

A.6.7 Ablation Study Description for the genetic algorithm

To better understand the contributions of each component in our genetic algorithm for feedback generation, we conduct an ablation study, systematically removing one component at a time:

1. Without Template Construction: The algorithm generates feedback without using the author-crafted issue-specific templates. This tests the importance of structured scaffolds in guiding precise and relevant feedback.

2. Without Plan Generation: The planner module is disabled, so the LLM does not generate

a knowledge-driven plan to enrich the template. Feedback relies only on the template and review segment.

3. Without Population Initialization: Multiple candidate feedbacks are not generated initially. Only a single feedback is produced per review segment, removing diversity from the initial population.

4. Without Fitness Evaluation: Candidate feedback is not scored or ranked. All generated feedback is treated equally, removing the model's ability to optimize for conciseness, relevance, and constructiveness.

5. Without Parent Selection & Crossover: Evolutionary steps are skipped, so no combination of high-quality candidates occurs. This evaluates the contribution of crossover in improving feedback quality and diversity.

6. Without Final Candidate Selection: The highest-scoring feedback is not selected at the end of the process. A candidate is chosen arbitrarily, which assesses the effect of selecting the optimal feedback. We present the results of the ablation study in Table 29. Each component contributes meaningfully to the quality of the generated feedback, highlighting the effectiveness of our approach.

A.6.8 Ablation on the Planner Component

We ablate the planner using multiple knowledge sources—abstract, reviewer-written strengths, and reviewer-written summaries—in Table 5. Our results show that incorporating all components is essential for generating the most specific feedback.

A.6.9 Ablation study on the rewards used in the fitness function for feedback generation

We present the results of the ablation study on different rewards in Table 4. We find that template adherence is the key factor driving the generation of highly effective feedback. Further analysis is provided in §5.

A.7 Example feedback

We show an example feedback generated by our approach in Fig 26.

A.8 Latency Analysis and Scaling

We show best and worst case estimate of deploying our feedback generation system to a real-world conference reviewing setup.

Assumptions:

- Average review: 5–10 segments
- LLM Q&A per batch of 16 segments: 1–2 s
- Segmentation: 0.1 s
- Issue Detection: 0.05 s
- Feedback generation: 1–2 s per review

Per-review Latency:

- LLM Q&A latency: 1–2 s
- Issue Detection latency: 0.05 s
- Feedback generation latency: 1–2 s
- Segmentation latency: 0.1 s

Total end-to-end latency per review: $0.1 + (1\text{–}2) + 0.05 + (1\text{–}2) \approx 2.15\text{–}4.15$ s per review

Scaling to 5,000 Reviews (Single GPU, Sequential):

- Best case: $5,000 \times 2.15\,\text{s} = 10,750\,\text{s} \approx 2.99$ hours
- Worst case: $5,000 \times 4.15\,\text{s} = 20,750\,\text{s} \approx 5.77$ hours

Multi-GPU Parallelization:

$$\text{Total wall-clock time} \approx \frac{\text{Sequential time}}{\text{Number of GPUs}}$$

By leveraging `vLLM` with a batch size of 16, LLM-based feature extraction is reduced to 1–2 seconds per review, and the genetic algorithm adds only 1–2 seconds, resulting in an end-to-end latency of ∼2–4 seconds per review. For a high-volume scenario of 5,000 reviews, sequential processing on a single GPU would require ∼3–6 hours, while parallelization across 4–8 GPUs reduces wall-clock time to ∼20–90 minutes, making the framework practical for deployment in time-sensitive contexts such as conference reviewing.

Issue / Heuristic	Template
H1. The results are not surprising	Your comment — "[insert reviewer comment here]" — suggests that the result is not surprising. While this may reflect your expertise, we encourage you to provide evidence or references (e.g., [references similar to title of the paper]) if this outcome is already known. What feels intuitive may not have been previously established, and empirical confirmation can still be a valuable contribution.
H2. The results contradict what I would expect	Your comment — "[insert reviewer comment here]" — indicates that the results contradict your expectations. Please consider that this may reflect confirmation bias, where prior beliefs influence interpretation. It's important to evaluate empirical findings objectively, even if they challenge existing assumptions. We encourage you to focus on the evidence presented and discuss how these findings advance understanding, rather than dismissing them based on expectation.
H3. The results are not novel	Your comment — "[insert reviewer comment here]" — raises concerns about novelty. If you believe the work duplicates prior work, please provide specific references ([references similar to the paper]) to support this claim (keeping in mind policies on concurrent work). If the paper's contribution is a reproduction, reimplementation, or analysis, please note that these are within the scope of the call and deserve fair consideration. Clear justification will help ensure the review is constructive and fair.
H4. This has no precedent in the existing literature	Your comment — "[insert reviewer comment here]" — indicates that the work has no clear precedent. While novelty is valuable, please be aware that highly novel contributions can be challenging to publish and may initially seem unfamiliar. Reviewers are encouraged to carefully consider the potential impact of innovative ideas rather than dismissing them due to lack of precedent. Providing constructive suggestions on how the authors can better frame or validate their novel contribution would be helpful.
H5. The results do not surpass the latest SOTA	Your comment — "[insert reviewer comment here]" — highlights that the results do not surpass the latest state-of-the-art (SOTA). Please keep in mind that achieving SOTA is neither necessary nor sufficient for a scientific contribution. Contributions improving other important aspects like efficiency, generalizability, interpretability, or fairness are also valuable. If the authors do not claim SOTA status, the lack of it should not be considered a weakness.
H6. The results are negative	Your comment — "[insert reviewer comment here]" — suggests the results are negative. Please consider that the bias toward publishing only positive results is a known issue across many fields. Negative or null results that show where methods systematically do not work are important for the community to understand limitations and avoid hype or overclaiming. Acknowledging the value of such findings can help make your review more balanced and constructive.
H7. This method is too simple	Your comment — "[insert reviewer comment here]" — suggests the method is too simple. However, the goal is to solve the problem effectively, not to add unnecessary complexity. Simpler methods often have advantages such as greater robustness, easier deployment, and better interpretability. Encouraging recognition of these benefits will help make your review more balanced and useful.
H8. The paper doesn't use [my preferred methodology]	Your comment — "[insert reviewer comment here]" — suggests a preference for a specific methodology (e.g., deep learning). Please note that NLP is an interdisciplinary field that values diverse contributions, including models, resources, surveys, data/linguistic/social analyses, theoretical work, and more. Limiting evaluation only to one methodology may overlook important advances. Recognizing this diversity will help make your review more comprehensive and fair.
H9. The topic is too niche	Your comment — "[insert reviewer comment here]" — suggests the topic is too niche. While the focus may be narrow, contributions to specific subfields can have significant impact and serve as important building blocks for the broader community. Recognizing the value of focused work helps ensure fair and balanced reviews.
H10. The approach is tested only on [not English]	Your comment — "[insert reviewer comment here]" — notes that the approach is tested only on [not English]. Please consider that monolingual research on any language is important both practically (developing methods and resources for that language) and theoretically (contributing to broader linguistic understanding). The NLP community values such focused contributions. Recognizing this helps ensure fair evaluation of research beyond English.
H11. The paper has language errors	Your comment — "[insert reviewer comment here]" — points out language errors. While clear writing is important, please prioritize the scientific content and contributions over minor language issues. Many papers can be improved with editorial help, but this should not overshadow the paper's value. Focusing on the core scientific merits will make your review more balanced and constructive.
H12. The paper is missing the [reference X]	Your comment — "[insert reviewer comment here]" — points out missing references. Please note ACL policy requires missing references to published highly relevant prior work that appeared at least three months before the submission deadline to be addressed seriously. Missing references to unpublished or very recent preprints should be treated as suggestions rather than critical flaws. Clarifying this distinction will help maintain fair and policy-compliant reviews.
H13. The authors could also do [extra experiment X]	Your comment — "[insert reviewer comment here]" — suggests additional experiments. While extra experiments can always be proposed, a paper only needs to provide sufficient evidence to support its claims. Additional experiments are generally considered "nice-to-have" and fit better in the suggestions section rather than as reasons for rejection. If you believe an extra experiment is essential for validity, please clearly justify why in your review.
H14. The authors should compare to a 'closed' model X	Your comment — "[insert reviewer comment here]" — requests comparisons to closed-source models (e.g., ChatGPT). Please consider that such comparisons are only meaningful if they directly support the paper's claims. Issues like test contamination and lack of transparency often limit their usefulness. Scientific progress often relies on openness and reproducibility, which closed models do not readily provide. Encouraging openness helps maintain rigorous evaluation standards.
H15. The authors should have done [X] instead	Your comment — "[insert reviewer comment here]" — suggests the authors should have done [X] instead. While multiple valid approaches often exist, this criticism is only warranted if the authors' choices prevent them from answering their research question, if their framing is misleading, or if the question itself lacks merit. Otherwise, such remarks are best framed as suggestions rather than weaknesses. Clarifying this distinction helps make your review fairer and more constructive.
H16. Limitations != weaknesses	Your comment — "[insert reviewer comment here]" — treats the paper's acknowledged limitations as weaknesses. While it's important to recognize limitations, they do not by themselves justify rejection. Most CL venues expect a Limitations section to promote transparency. If you believe a limitation seriously undermines the work, please clearly explain why. Distinguishing limitations from fatal flaws will improve the fairness and usefulness of your review.
The paper is missing relevant references	Your comment — "[insert reviewer comment here]" — needs to be rewritten by mentioning the exact references the authors are missing.
X is not clear	Your comment — "[insert reviewer comment here]" — needs to be rewritten by specifying the details that are exactly missing such as Y and Z are missing from the description of X.
The formulation of X is wrong	Your comment — "[insert reviewer comment here]" — needs to be rephrased as the formulation of X misses the component Y.
The contribution is not novel	Your comment — "[insert reviewer comment here]" — needs to be rephrased as highly similar work X and Y has been published 3+ months prior to submission deadline.
The paper is missing recent baselines	Your comment — "[insert reviewer comment here]" — needs to be rephrased as the proposed method should be compared against recent methods X, Y and Z.
X was done in the way Y	Your comment — "[insert reviewer comment here]" — needs to be rephrased as X was done in the way Y which has the disadvantage Z.

Table 12: Templates corresponding to heuristics and common reviewer issue types.

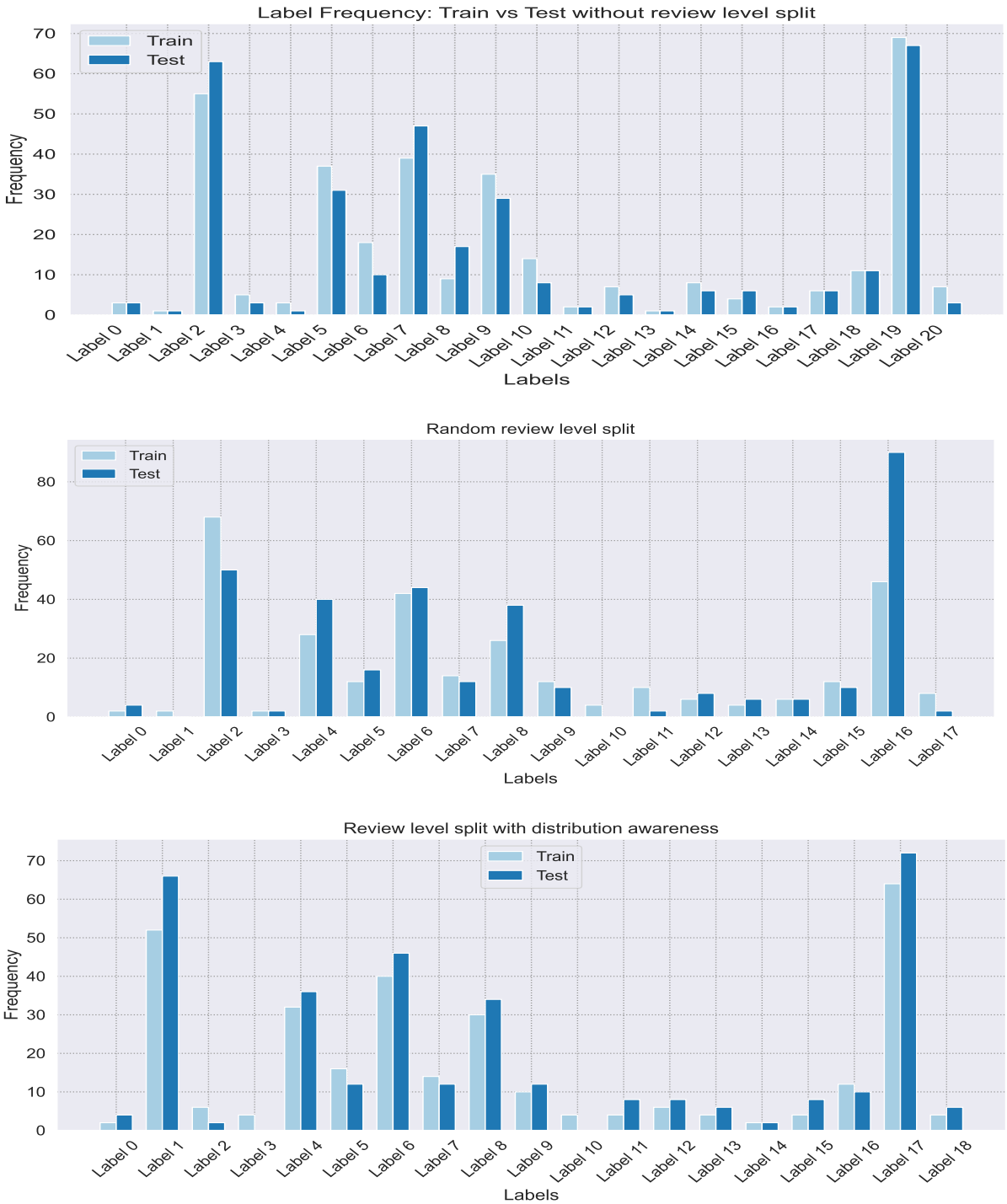
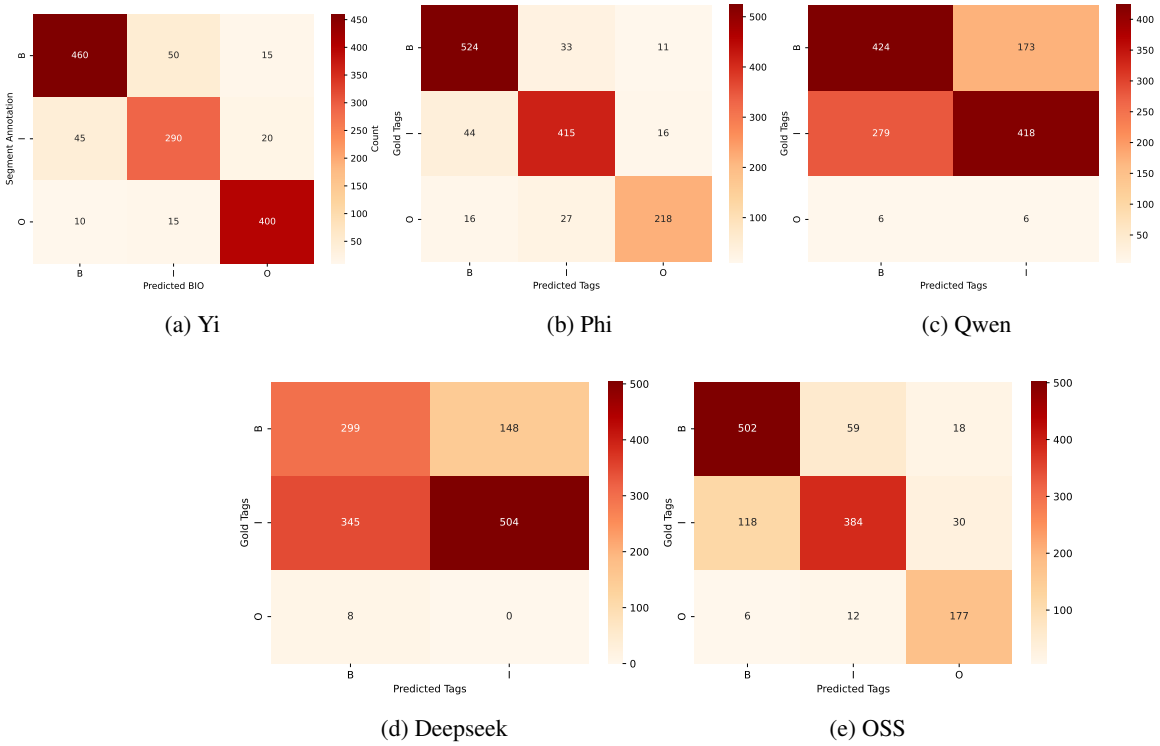


Figure 7: Comparing the label distribution using various split methods.



(a) Yi (b) Phi (c) Qwen

(d) Deepseek (e) OSS

Figure 8: Confusion matrices for all the models on the **segment detection** task.

Issue Identification Prompt Library

Feature Extraction Prompt

You are given: - An issue label: issue - Why this issue is problematic: problem - Review segments (from NLP paper reviews): segments - Desired number of questions: 10
Your goal: Extract abstract, high level features that determine whether each review segment reflects the specified issue. Express each feature as an objective Yes/No question.
Rules you MUST follow: 1) Use ONLY the provided segments. Do NOT rely on external knowledge or assumptions. 2) Work at the ABSTRACT level (features that could generalize across wording), not surface phrasing. 3) Every question must be answerable by inspecting a single segment in isolation. 4) Questions must be neutral (no leading language), atomic (one idea per question), and non-redundant. 5) Avoid double negatives, multi-part questions, subjective terms ("clearly", "obviously"), or jargon unless it appears in the segments. 6) Prefer beginnings like "Does...", "Is...", "Are...", "Has...", "Do...", "Can...". 7) Keep each question concise (ideally 18 words). 8) Produce EXACTLY N questions. If N is not given, produce 10. 9) OUTPUT FORMAT: Return ONLY a Python list of strings, with double quotes, no code fences, no comments, no trailing commas. 10) Do NOT include any analysis, notes, or explanations in the output — only the list.
Step-by-step procedure: A) Parse and inventory: - Scan the segments to note common patterns about claims, evidence, specificity, comparisons, citations, quantification, assumptions, scope/coverage, and consistency. - Identify cues that would indicate presence/absence of the issue issue.
B) Abstract feature mining: - Convert recurring patterns into abstract properties that can reflect the similarity between the review segments you are given - Ensure each property directly supports diagnosing issue as problematic because problem.
C) Draft discriminative Yes/No tests: - For each property, write a Yes/No question that a reviewer could answer from a single segment. - Ensure questions collectively cover: evidence/citations, specificity/locating, correctness/faithfulness, scope, quantification, reproducibility/actionability, internal consistency, and fairness/balance (as applicable).
D) Prune and refine: - Remove overlaps; keep the most general, segment-checkable forms. - Rewrite to be atomic, neutral, and concise. - Ensure each question distinguishes "issue present" vs "issue absent".
E) Final checks (must pass all): - [ ] Exactly N questions. - [ ] All are Yes/No answerable from a single segment. - [ ] No duplicates or near-duplicates. - [ ] No restating issue verbatim; focus on testable properties. - [ ] Python list of strings ONLY, no extra text.
Now produce the final output.

Feature QA prompt

You will be given a review segment. Your task is to evaluate its quality by answering a series of Yes/No questions.
Review Segment: review
Question: question
Respond strictly with either:
[[Yes]] if the answer is Yes
[[No]] if the answer is No
[[Other]] if the question is irrelevant

Figure 9: Issue Identification Prompt

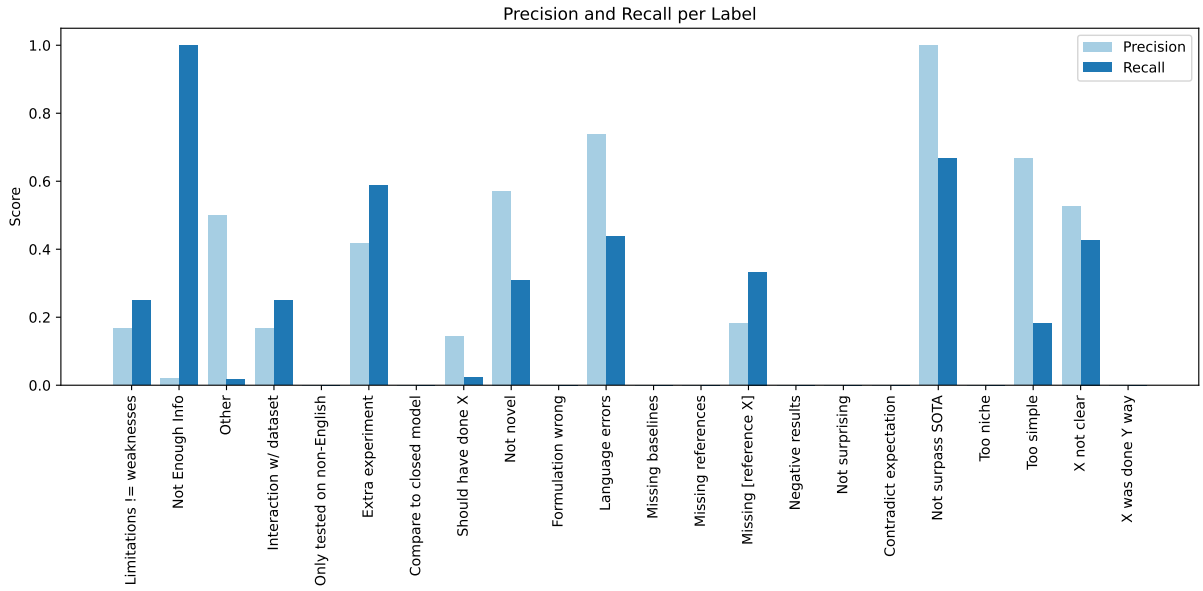


Figure 10: Zero shot results for **issue detection** with GPT OSS 20B. 10 out of 22 classes remain undetected yielding a miss rate of $10/22 = 0.454 \sim 45\%$.

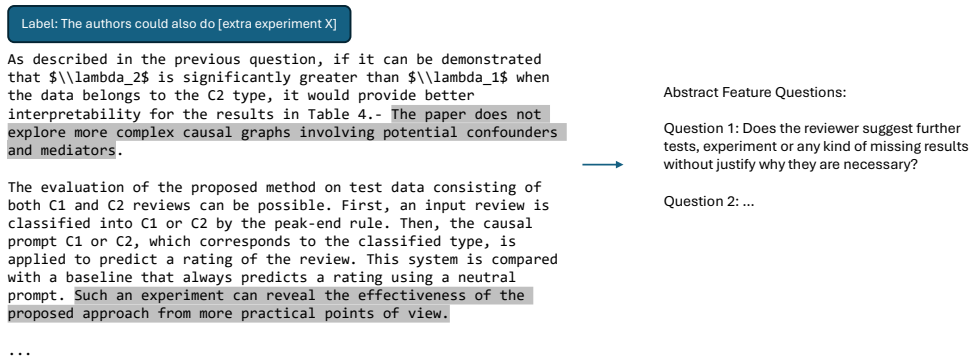


Figure 11: Example of abstract feature questions generated by our **issue detection** approach.

Method	Yi	Phi	Qwen	Deep.	Oss.
LLM_{Zero}	0.04	0.05	0.06	0.03	0.23
LLM_{Fine}	0.23	0.12	0.13	0.11	0.19
LLM_{QA}	0.05	0.12	0.17	0.20	0.12
$\text{LLM}_{\text{QAFine}}$	0.10	0.10	0.07	0.08	0.11
RoBERTa			0.39		
KNN	0.34	0.38	0.39	0.38	0.37
LogReg-L2	0.32	0.37	0.31	0.31	0.33
LogReg-L1	0.32	0.38	0.35	0.32	0.32
RF	0.34	0.41	0.40	0.35	0.33
DT	0.26	0.27	0.24	0.22	0.25
SVM-RBF	0.34	0.35	0.35	0.34	0.32
SVM-Lin	0.41	0.42	0.38	0.35	0.34
GBoost	0.46	0.49	0.43	0.36	0.41
AdaB	0.43	0.48	0.36	0.36	0.37
ExtraT	**0.51**	**0.51**	**0.44**	**0.42**	**0.44**
MLP	0.45	0.49	0.41	0.42	0.41

Table 13: Cross-Validation Performance comparison of methods for the **issue detection** task across LLMs and the encoder-only baseline (RoBERTa) based on F0.5 scores. The best performing classifier is in bold with the best performing LLM highlighted.

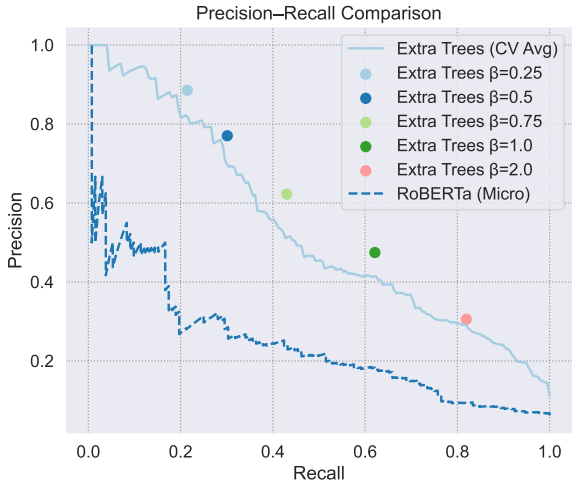


Figure 12: Precision-Recall Curve for RoBERTa and our approach operating at different thresholds.

	Review-level-random-split	Review-level-split	sklearn-split
Distance	0.036	0.014	0.015

Table 14: Distance value for the three different split methods using evenly split 50% of data

Method	Yi		Phi		Qwen		Deep.		Oss.	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
KNN	0.51	0.27	0.43	0.23	0.44	0.21	0.41	0.18	0.44	0.20
Logistic Regression (L2)	0.52	0.32	0.52	0.30	0.46	0.33	0.39	0.29	0.41	0.28
Logistic Regression (L1)	0.49	0.35	0.46	0.34	0.40	0.34	0.42	0.31	0.42	0.32
Random Forest	0.78	0.19	0.79	0.21	0.74	0.16	0.66	0.15	0.65	0.18
Decision Tree	0.36	0.38	0.34	0.37	0.33	0.36	0.25	0.26	0.32	0.34
SVM (RBF)	0.83	0.15	0.79	0.19	0.69	0.12	0.77	0.13	0.66	0.12
SVM (Linear)	0.43	0.34	0.45	0.33	0.39	0.34	0.36	0.33	0.34	0.31
Gradient Boosting	0.52	0.31	0.54	0.37	0.49	0.30	0.41	0.23	0.46	0.29
AdaBoost	0.47	0.33	0.52	0.38	0.39	0.29	0.39	0.29	0.40	0.29
Extra Trees	0.76	0.23	0.75	0.23	0.70	0.18	0.68	0.17	0.64	0.19
MLP	0.51	0.31	0.56	0.33	0.44	0.33	0.49	0.28	0.47	0.27
Gaussian Naive Bayes	0.23	0.47	0.29	0.52	0.22	0.54	0.23	0.57	0.24	0.48

Table 15: Precision and Recall of different LLMs using our approach for **issue detection**.

Review Extension

Original Comment: "...no check for false negatives… only one language, one domain, one genre… no significance testing." **Issue Detected**: **The authors could also do extra experiment, X**
Generated feedback:
You can rewrite your comment as the follows: While the paper presents a promising benchmark and methodology, the evaluation appears limited in scope. The experiments focus solely on English data from a single domain and genre, which may restrict the generalizability of the results. Including experiments across multiple languages or domains could strengthen the empirical claims. Additionally, incorporating significance testing would help confirm that the observed improvements are statistically meaningful. Overall, expanding the evaluation would improve both the rigor and the impact of the work.

Figure 13: Review Extension: Generating full reviews instead of feedback

Role Separation (Author-Facing Feedback)

Original Comment: "...no check for false negatives… only one language, one domain, one genre… no significance testing."
Issue Detected: **The authors could also do extra experiment, X**
Generated feedback:
The current experiments demonstrate the benchmark and methodology effectively, but they are limited to English, a single domain, and one genre. To strengthen the claims, consider including additional languages, domains, or genres where feasible. Also, performing significance testing would provide stronger evidence for the reported improvements. While these are suggestions to enhance the work, the main contribution remains valid and valuable within the presented scope.

Figure 14: Role Separation: Generating feedback to authors instead of reviewers

Method	Model	Const.	Relev.	Spec.	Conc.
1-pass	Yi	1.9	2.0	1.8	1.9
	Phi	2.1	2.2	2.0	2.1
	Qwen	1.8	1.9	1.7	1.8
	Deep.	1.7	1.8	1.6	1.7
	Oss.	1.9	2.0	1.8	1.9
BoN	Yi	2.0	2.2	2.4	2.3
	Phi	2.2	2.3	2.1	2.2
	Qwen	2.0	2.1	2.0	2.0
	Deep.	1.8	1.9	1.7	1.8
	Oss.	2.0	2.1	2.0	2.1
Self-Ref.	Yi	2.3	2.4	2.4	2.5
	Phi	2.4	2.6	2.8	2.9
	Qwen	2.2	2.4	2.4	2.7
	Deep.	2.4	2.4	2.5	2.3
	Oss.	3.0	3.1	2.9	3.0
Temp	Yi	2.8	2.9	2.7	2.8
	Phi	3.1	3.2	3.0	3.3
	Qwen	2.9	3.0	2.8	2.9
	Deep.	2.7	2.8	2.6	2.8
	Oss.	2.8	2.9	2.7	2.8
Plan	Yi	3.0	3.1	2.9	3.0
	Phi	3.3	3.4	3.2	3.5
	Qwen	3.1	3.2	3.0	3.1
	Deep.	2.9	3.0	2.8	3.0
	Oss.	3.0	3.1	2.9	3.0
Ours	Yi	3.9	3.8	3.8	3.8
	Phi	**4.3**	**4.3**	**4.2**	**4.3**
	Qwen	3.8	3.8	3.7	3.7
	Deep.	3.5	3.6	3.6	3.5
	Oss.	4.0	4.1	4.0	3.8

Table 16: Performance comparison of various models on the **feedback generation** task using four customized **automated metrics** from Prometheus V2: Constructiveness (Const.), Relevance (Relev.), Specificity (Spec.), and Conciseness (Conc.). Overall best results are **bolded** and method-wise best results are underlined.

Classifier	F0.5
KNN	0.28
Logistic Regression (L2)	0.39
Logistic Regression (L1)	0.39
Random Forest	0.10
Decision Tree	0.21
SVM (RBF)	0.22
SVM (Linear)	0.37
SVM (Poly)	0.00
AdaBoost	0.25
Extra Trees	0.09
Multi-layer Perceptron (MLP)	0.30
Gaussian Naive Bayes	0.34

Table 17: Performance of various classifiers on the **issue detection** task using representations obtained from all-MiniLM-L6-v2 sentence transformer model.

Classifier	F0.5	Precision	Recall
KNN	0.34	0.41	0.20
Logistic Regression (L2)	0.37	0.40	0.30
Logistic Regression (L1)	0.39	0.42	0.32
Random Forest	0.42	0.67	0.17
Decision Tree	0.34	0.33	0.38
SVM (RBF)	0.36	0.66	0.13
SVM (Linear)	0.34	0.34	0.34
SVM (Poly)	0.08	0.49	0.02
AdaBoost	0.43	0.47	0.32
Extra Trees	0.39	0.61	0.16
MLP	0.42	0.47	0.30
Gaussian Naive Bayes	0.22	0.19	0.57

Table 18: Performance of various classifiers using Phi-4 generated 5 question feature vector for the **issue detection** task.

Methods	Ours. train	Ours. test	Emb. train	Emb. test
KNN	0.36	0.37	0.40	0.19
Logistic Regression (L2)	0.42	0.43	0.47	0.18
Logistic Regression (L1)	0.38	0.41	0.42	0.18
Random Forest	0.40	0.45	0.40	0.22
Decision Tree	0.34	0.37	0.28	0.19
SVM (RBF)	0.25	0.39	0.46	0.19
SVM (Linear)	0.36	0.40	0.45	0.17
SVM (Poly)	0.02	0.09	0.34	0.32
Gradient Boosting	0.36	0.41	0.33	0.21
AdaBoost	0.42	0.39	0.33	0.17
Extra Trees	0.37	0.44	0.46	0.28
Multi-layer Perceptron (MLP)	0.39	0.40	0.43	0.18
Gaussian Naive Bayes	0.30	0.31	0.42	0.18
Stochastic Gradient Descent (SGD)	0.35	0.43	0.38	0.20

Table 19: F0.5 score on even split of training and test with 50% each for **issue detection**. Embedding (Emb.) corresponds to representations obtained from `all-MiniLM-L6-v2` sentence transformer model.

Model	0.25			0.5			0.75			1.0			2.0		
	$F_{0.25}$	P	R	$F_{0.50}$	P	R	$F_{0.75}$	P	R	$F_{1.0}$	P	R	$F_{2.0}$	P	R
KNN	0.56	0.81	0.12	0.44	0.58	0.23	0.39	0.51	0.31	0.37	0.37	0.4	0.5	0.2	0.6
Logistic Regression (L2)	0.58	0.64	0.26	0.50	0.59	0.33	0.46	0.49	0.46	0.46	0.46	0.5	0.5	0.3	0.6
Logistic Regression (L1)	0.65	0.81	0.19	0.52	0.65	0.32	0.48	0.55	0.39	0.46	0.52	0.4	0.5	0.3	0.6
Random Forest	0.74	0.86	0.23	0.61	0.76	0.35	0.55	0.64	0.44	0.53	0.59	0.5	0.6	0.4	0.7
Decision Tree	0.35	0.35	0.38	0.35	0.35	0.38	0.36	0.35	0.38	0.36	0.35	0.4	0.4	0.3	0.4
SVM (RBF)	0.72	0.90	0.17	0.56	0.67	0.33	0.50	0.62	0.38	0.49	0.51	0.5	0.6	0.3	0.7
SVM (Linear)	0.54	0.74	0.15	0.45	0.51	0.33	0.43	0.44	0.42	0.43	0.39	0.5	0.5	0.3	0.7
Gradient Boosting	0.61	0.69	0.24	0.53	0.60	0.36	0.50	0.53	0.46	0.49	0.49	0.5	0.6	0.4	0.6
AdaBoost	0.59	0.66	0.28	0.51	0.59	0.33	0.47	0.51	0.44	0.48	0.44	0.6	0.5	0.3	0.7
Extra Trees	0.75	0.90	0.24	0.62	0.76	0.39	0.57	0.70	0.44	0.54	0.65	0.5	0.6	0.4	0.7
Multi-layer Perceptron (MLP)	0.67	0.78	0.22	0.56	0.66	0.36	0.50	0.64	0.38	0.47	0.57	0.4	0.5	0.3	0.6

Table 20: Performance of different models across thresholds 0.25, 0.5, 0.75, 1.0, and 2.0. Metrics shown are $F_{threshold}$ (e.g., $F_{0.25}$ for threshold 0.25), P (Precision), and R (Recall).

Baselines	Yi		Phi		Qwen		Deep.		Oss.	
	P	R	P	R	P	R	P	R	P	R
$\texttt{LLM}_{\texttt{zero}}$	0.03	0.12	0.09	0.11	0.06	0.10	0.03	0.10	0.22	0.24
$\texttt{LLM}_{\texttt{Fine}}$	0.23	0.23	0.11	0.18	0.13	0.17	0.10	0.19	0.19	0.21
$\texttt{LLM}_{\texttt{QA}}$	0.10	0.06	0.10	0.28	0.16	0.22	0.19	0.26	0.10	0.40
$\texttt{LLM}_{\texttt{QAFine}}$	0.10	0.20	0.09	0.27	0.05	0.12	0.07	0.19	0.10	0.17

Table 21: Precision (P) and Recall (R) of various LLMs for the **issue detection** task.

Conciseness Evaluation Prompt

Task Description: You are given an instruction, a response to evaluate, and a score rubric representing an evaluation criterion.
Your goal: 1. Write detailed feedback that assesses the response strictly based on the rubric. 2. Assign a score between 1 and 5, referring to the rubric. 3. Output in the following format:

> Feedback: (write a feedback for criteria) [RESULT] (an integer number between 1 and 5)

4. Do not include any opening, closing, or explanations.
Instruction to evaluate: You are an expert evaluating feedback generated for improving review segments. The feedback provided will help the reviewer improve the segment.
Weakness: weakness
Feedback to evaluate: feedback
Score Rubric: [Does the feedback communicate suggestions concisely? Feedback should be brief, precise, and to the point, avoiding unnecessary verbosity while remaining actionable.]

- Score 1: Feedback is wordy, repetitive, or confusing; hard to extract actionable guidance.
- Score 2: Feedback conveys ideas but is often verbose, vague, or partially actionable; contains unnecessary wording.
- Score 3: Feedback is moderately concise, understandable, and partially actionable; some redundancy or extra wording may remain.
- Score 4: Feedback is clear, focused, and mostly concise; communicates actionable guidance efficiently, with minor verbosity.
- Score 5: Feedback is precise, concise, and easy to interpret; provides direct, actionable suggestions covering both minor and substantive issues effectively.

Feedback:

Figure 15: `Prometheus` Conciseness Prompt

Relevance Evaluation Prompt

Task Description: You are given an instruction, a response to evaluate, and a score rubric representing an evaluation criterion.
Your goal: 1. Write detailed feedback that assesses the response strictly based on the rubric. 2. Assign a score between 1 and 5, referring to the rubric. 3. Output in the following format:

> Feedback: (write a feedback for criteria) [RESULT] (an integer number between 1 and 5)

4. Do not include any opening, closing, or explanations.
Instruction to evaluate: You are an expert evaluating feedback generated for improving review segments. The feedback provided will help the reviewer improve the segment.
Weakness: weakness
Feedback to evaluate: feedback
Score Rubric: [Is the feedback directly related to the content or structure of the review? Feedback should focus on aspects that help improve the review, including conciseness, reasoning, accuracy, or actionable corrections.]

- Score 1: Feedback is off-topic or unrelated to the review's content or structure; provides no actionable guidance.
- Score 2: Feedback mentions the review but mostly addresses minor or tangential points; limited actionable guidance.
- Score 3: Feedback is partially relevant; identifies some issues but mixes relevant and loosely connected commentary; limited guidance.
- Score 4: Feedback is clearly related to review content or structure; provides actionable suggestions addressing key issues.
- Score 5: Feedback is tightly focused on review content or structure; comments are precise, actionable, and justified, improving clarity, reasoning, or accuracy.

Feedback:

Figure 16: `Prometheus` Relevance Prompt

Constructiveness Evaluation Prompt

Task Description: You are given an instruction, a response to evaluate, and a score rubric representing an evaluation criterion.
Your goal: 1. Write detailed feedback that assesses the response strictly based on the rubric. 2. Assign a score between 1 and 5, referring to the rubric. 3. Output in the following format:

> Feedback: (write a feedback for criteria) [RESULT] (an integer number between 1 and 5)

4. Do not include any opening, closing, or explanations.
Instruction to evaluate: You are an expert evaluating feedback generated for improving review segments. The feedback provided will help the reviewer improve the segment.
Weakness: weakness
Feedback to evaluate: feedback
Score Rubric: [Does the feedback offer suggestions for how the reviewer can improve? Constructive feedback should guide revisions, not just point out flaws, and can address both minor textual issues and substantive content.]

- Score 1: Feedback only identifies flaws without suggesting improvements; unhelpful or dismissive.
- Score 2: Feedback includes vague or superficial advice; little actionable guidance.
- Score 3: Feedback identifies issues and offers some guidance but partially unclear or limited.
- Score 4: Feedback provides clear and relevant suggestions; mostly actionable, may lack some detail.
- Score 5: Feedback consistently identifies issues and provides specific, practical, and helpful suggestions; supports targeted improvements effectively.

Feedback:

Figure 17: `Prometheus` Constructiveness Prompt

Specificity Evaluation Prompt

Task Description: You are given an instruction, a response to evaluate, and a score rubric representing an evaluation criterion.
Your goal: 1. Write detailed feedback that assesses the response strictly based on the rubric. 2. Assign a score between 1 and 5, referring to the rubric. 3. Output in the following format:

> Feedback: (write a feedback for criteria) [RESULT] (an integer number between 1 and 5)

4. Do not include any opening, closing, or explanations.
Instruction to evaluate: You are an expert evaluating feedback generated for improving review segments. The feedback provided will help the reviewer improve the segment.
Weakness: weakness
Feedback to evaluate: feedback
Score Rubric: [Does the feedback refer to specific parts or issues in the review? Feedback should clearly indicate what needs improvement, avoiding vague statements without context.]

- Score 1: Feedback is entirely vague or generic; no reference to particular parts of the review; not actionable.
- Score 2: Feedback hints at an issue but lacks concrete references; reviewer cannot easily locate the problem.
- Score 3: Feedback identifies general areas or sections but does not pinpoint exact sentences or claims; partially actionable.
- Score 4: Feedback refers to specific sections or issues; minor vagueness may remain; mostly actionable.
- Score 5: Feedback clearly identifies exact parts of the review and provides precise, actionable guidance, including minor textual or deeper content issues when justified.

Feedback:

Figure 18: `Prometheus` Specificity Prompt

Lazy Thinking Issues

H1. The results are not surprising: Many findings seem obvious in retrospect, but this does not mean the community is already aware of them. Some findings may seem intuitive but haven't been empirically tested.
H2. The results contradict expectations: Reviewer may be biased against findings that challenge prior beliefs.
H3. The results are not novel: If a claimed novel method resembles existing work, provide references. Reproduction, reimplementation, or analysis papers may still be in scope.
H4. No precedent in existing literature: Novel papers are harder to publish; reviewers may be overly conservative.
H5. Results do not surpass latest SOTA: SOTA results are not required; other contributions (efficiency, fairness, interpretability) matter.
H6. Negative results: Publishing negative results is important; systematic failures provide valuable knowledge.
H7. Method too simple: Simple solutions are often preferable—less brittle and easier to deploy.
H8. Missing preferred methodology: NLP is interdisciplinary; multiple approaches (models, resources, analyses) are valid.
H9. Topic too niche: Significant contributions can be made to narrow subfields.
H10. Tested only on non-English: Monolingual studies are important for both practical and theoretical contributions.
H11. Language errors: Focus on scientific content; minor writing issues should not dominate.
H12. Missing reference(s): Prior work published >3 months before submission should be cited; preprints are optional.
H13. Extra experiments suggested: Optional experiments are "nice-to-have" and belong in suggestions, not as reasons to reject.
H14. Compare to closed model: Only meaningful if it directly affects the paper's claim; closed models often introduce methodological issues.
H15. Authors should have done X instead: Only relevant if choices prevent addressing the research question; otherwise a suggestion, not a weakness.
H16. Limitations != weaknesses: Limitations sections are standard; do not equate limitations with reasons to reject unless strongly justified.

Specificity Issues

Specificity Issues:
Missing relevant references: The review fails to mention key prior work (e.g., XYZ).
X is unclear: Important details (e.g., Y and Z) are missing from description.
Formulation of X is wrong: The definition or equation of X omits critical factors (e.g., Y).
Contribution not novel: Similar work (e.g., X, Y) has been published >3 months prior.
Missing recent baselines: Proposed methods should be compared to recent approaches X, Y, Z.
X done in way Y: Implementation choices lead to known limitations (e.g., disadvantage Z).

Task

Task: You are an expert at providing actionable feedback to improve a review segment. Given a review, the segment, and lazy thinking annotation, provide feedback that the reviewer can use to revise the segment.

Review Segment: review segment
Issue: issue
Feedback :

Figure 19: Zero shot Feedback generation prompt

Templatic Feedback Generation Prompt

Task: You are an expert at providing actionable feedback to improve the writing of review segments. Given a review segment and its identified issue, your goal is to produce feedback that the reviewer can use to revise the segment. You are also provided a **feedback template** for the identified issue.
Instructions: Adapt the provided feedback template to match the comment in the review segment. Ensure the feedback is actionable, specific, and targeted to improving the segment.
Review Segment: weakness
Identified Issue: identified issue
Feedback Template: template

Output: The feedback generated for this review segment, following the guidance of the template.

Figure 20: Prompt for Templatic Feedback Generation using Identified Issues.

Planner Prompt for Feedback Generation

System Instruction: You are a planning agent for feedback generation.
Input:

- **Review Comment:** {review}
- **Issue:** {issue}
- **Abstract of the paper:** {abstract}
- **Summary written by reviewer:** {summary}
- **Strengths written by reviewer:** {strengths}

Task:

1. For each label, identify which external knowledge (Abstract, Summary, Template) is needed based on the mapping.
2. Suggest how to use the available knowledge (e.g., quote from abstract, summarize key claim).
3. Provide an explanation for your choice of plan.

Output: A JSON list with the following structure:

- `plan`: brief instructions on how to use the knowledge
- `explanation`: reasoning behind the chosen plan

Example Output:

```
[
  {
    "plan": "Use summary to show that results differ from intuition",
   "explanation": "The summary provides key claims that can highlight why the results are surprising."
  },
  {
    "plan": "Use abstract + summary to check novelty",
   "explanation": "Both abstract and summary help determine whether the contribution is novel."
  }
]
```

Figure 21: Full Planner Prompt for Feedback Generation.

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Variance
$F_{0.5}$	0.55	0.55	0.42	0.52	0.56	0.002
Precision	0.74	0.88	0.64	0.69	0.81	0.007
Recall	0.27	0.22	0.18	0.26	0.25	0.0009

Table 22: Cross-validation results over 5 folds showing $F_{0.5}$, Precision, Recall, and their variances.

Model	Mean $F_{0.5}$	Mean Precision	Mean Recall
KNN	0.544	0.667	0.314
Logistic Regression (L2)	0.514	0.576	0.365
Logistic Regression (L1)	0.456	0.477	0.391
Random Forest	0.554	0.761	0.270
SVM (RBF)	0.463	0.948	0.153
Extra Trees	0.575	0.783	0.281

Table 23: Mean $F_{0.5}$, precision, and recall across different models using human-written questions as features.

Feedback Generation Prompt

System: You are an expert at generating actionable feedback to improve review segments.
Instructions: Given: 1. A review segment (weakness) 2. Identified issue(s) 3. A feedback template for the issue(s) 4. A plan describing what to use as knowledge sources (abstract, summary, template) to guide feedback. 5. An explanation describing how to use the knowledge sources.
Your task: - Generate actionable feedback for the reviewer to improve the review segment. - Incorporate the plan into the template: adapt the template to match the specific review segment and issue. - Keep the feedback precise, relevant, and constructive. - Output only the feedback text, do not include any explanations or extra text.
Review Segment: weakness
- Identified Issue: identified issue
Feedback Template: template
Plan: plan **Explanation**: explanation
Example Output: "Feedback: The reviewer should clarify the novelty of the contribution by referencing prior work; following the template, highlight which claims are incremental and which are novel. [5]"

Figure 22: Feedback generation using Plan and Template

Self-Refine Feedback Generation Prompt

System: You are an expert at generating and refining actionable feedback for improving review segments.
Instructions: Given: 1. A review segment (weakness) 2. Identified issue(s) 3. An initial draft feedback
Your task: - Generate detailed feedback that helps the reviewer improve the review segment. - Critically evaluate the initial feedback and refine it to be more precise, constructive, and actionable. - Ensure the feedback is relevant to the identified issue(s) and avoids vague statements. - Output only the refined feedback text, do not include explanations, reasoning, or any extra text.
Variables: - Review Segment: weakness - Identified Issue: identified issue - Initial Feedback: initial feedback
Example Output: "Feedback: The reviewer should provide concrete examples illustrating missing baselines; clearly indicate which comparisons are necessary and why. [5]"

Figure 23: Self-refine feedback generation for review segments

Cross-Over Feedback Prompt (5 Parents)

System: You are an expert at generating high-quality feedback for review segments by synthesizing multiple sources.
Instructions: Given: 1. A review segment (weakness) 2. Identified issue(s) 3. Five parent feedback drafts
Your task: - Combine the key insights from all 5 parent feedbacks into a single, coherent feedback. - Ensure that the final feedback is precise, constructive, and actionable. - Retain the strongest points from each parent, remove redundancy, and resolve conflicting suggestions. - Feedback must directly address the identified issue(s) and help the reviewer improve the segment. - Output only the final synthesized feedback, no explanations or commentary.
Variables: - Review Segment: weakness - Identified Issue: identified issue - Parent Feedbacks: 1) parent1 2) parent2 3) parent3 4) parent4 5) parent5
Example Output: "Feedback: The reviewer should clarify the novelty of the contribution and reference prior work where necessary; highlight incremental contributions while maintaining clarity and specificity. [5]"

Figure 24: Cross-over feedback synthesis for 5 parent feedbacks

Model	Constructiveness		Relevance		Specificity		Conciseness	
	Kendall τ	Spearman ρ	Kendall τ	Spearman ρ	Kendall τ	Spearman ρ	Kendall τ	Spearman ρ
Yi	0.75	0.83	0.76	0.84	0.75	0.83	0.76	0.85
Phi	0.78	0.88	0.77	0.87	0.76	0.86	0.78	0.88
Qwen	0.76	0.84	0.75	0.85	0.76	0.84	0.75	0.84
Deep.	0.75	0.83	0.75	0.84	0.75	0.83	0.75	0.83
Oss.	0.77	0.85	0.76	0.85	0.76	0.85	0.77	0.86
Average	0.76	0.85	0.76	0.85	0.76	0.84	0.76	0.85

Table 24: Kendall τ and Spearman ρ correlations with `Prometheus` for different models across four metrics: Constructiveness, Relevance, Specificity, and Conciseness. Method-wise best correlations are underlined. The last row shows the average correlation across all models and metrics.

	Constr.			Rel.			Spec.			Conc.		
Model	Pr-v1	GPT-4o	Pr-v2	Pr-v1	GPT-4o	Pr-v2	Pr-v1	GPT-4o	Pr-v2	Pr-v1	GPT-4o	Pr-v2
Yi	0.73	0.76	**0.75**	0.74	0.75	**0.76**	0.73	0.74	**0.75**	0.74	0.75	**0.76**
Phi	0.77	0.78	**0.78**	0.76	0.77	**0.77**	0.75	0.76	**0.76**	0.77	0.77	**0.78**
Qwen	0.75	0.75	**0.76**	0.74	0.75	**0.75**	0.75	0.75	**0.76**	0.74	0.75	**0.75**
Deep.	0.74	0.74	**0.75**	0.74	0.74	**0.75**	0.74	0.74	**0.75**	0.74	0.74	**0.75**
Oss.	0.76	0.76	**0.77**	0.75	0.75	**0.76**	0.75	0.75	**0.76**	0.76	0.76	**0.77**
Avg	0.75	0.76	**0.76**	0.75	0.76	**0.76**	0.74	0.75	**0.76**	0.75	0.76	**0.76**

Table 25: Kendall τ correlations across review quality dimensions for `Prometheus` v1 (Pr-v1), v2 (Pr-v2) and GPT-4o respectively.

Model	Review Ext. (%)	Role Sep. (%)
Yi	42	23
DeepSeek	47	30
OSS	43	28
Qwen	45	25
Phi	**41**	**23**

Table 26: Comparison of review generation issues across models. Review Extension refers to instances where the LLM generates full reviews instead of targeted feedback, while Role Separation indicates cases where feedback is misdirected to authors instead of reviewers. The least of the issues is highlighted

Hyper-parameter	Description	Value / Setting
T	Number of author-crafted templates	25
n	Number of initial candidate feedbacks per segment	10
n_{parents}	Number of parents selected per generation for crossover	5
τ	Temperature parameter for Boltzmann selection	0.1
$n_{\text{generations}}$	Number of evolutionary generations	3
n_{sent}	Number of maximum sentences allowed	5

Table 27: Hyper-parameters used in the genetic algorithm for **feedback generation** algorithm.

Review Extension and Role Separation Detection Prompt

System: You are an expert at analyzing peer review feedback.
Instructions: Given a review, your task is to classify each review comment according to two phenomena:
1. **Review Extension** – The feedback generates a full review instead of targeted feedback for a specific review segment. This includes writing additional evaluations, repeating comments, or elaborating beyond what is necessary.
2. **Role Separation** – The feedback provides suggestions meant for the author rather than the reviewer. Comments may directly address the author, suggest rewriting the review for the author, or cross the boundary between reviewing and authoring.
Task: - For each review comment, indicate whether it exhibits Review Extension, Role Separation, both, or neither. - Provide a brief justification for your classification.
Variables: - Review Comment: review
Output Format: - JSON array with fields:

```
[
  {
    "comment": "text of the review comment",
    "review_extension": true/false,
    "role_separation": true/false,
    "justification": "brief explanation"
  },
  ...
]
```

Figure 25: Prompt for identifying Review Extension and Role Separation in peer review comments.

Example Feedback: Extra Experiments

Issue Detected: The authors could also do [extra experiment X]
Reviewer Comment: "...no check for false negatives... only one language, one domain, one genre... no significance testing"
Feedback: Your comment highlights suggestions for additional experiments and concerns about generalization. While extra experiments (e.g., multilingual evaluation or statistical tests) are always possible, the paper provides sufficient evidence to support its main claims. The contribution introduces the DYNAMICER benchmark and a temporal segmented clustering method that improves retrieval-augmented generation for emerging mentions. These contributions are demonstrated convincingly within the chosen scope. Monolingual studies (English, sports domain, social media genre) are still valuable practically and theoretically. If you believe significance testing or cross-domain validation is essential for validity, please clearly justify why; otherwise, these points are better framed as suggestions rather than reasons for rejection.

Figure 26: Feedback generated from our approach on an instance of our dataset, LAZYREVIEWPLUS

Method	Const.	Relev.	Spec.	Conc.
1-pass	1.8	1.9	2.2	2.2
BoN	2.4	2.4	2.5	2.5
Self-Ref.	2.5	2.6	2.6	2.6
Temp	2.8	2.7	2.8	2.8
Plan	3.2	3.1	3.2	3.2
Ours	**4.1**	**4.2**	**4.0**	**4.3**

Table 28: Human Evaluation of the best performing model Phi on the **feedback generation** task using Constructiveness (Const.), Relevance (Relev.), Specificity (Spec.) and Conciseness (Conc.)

Model	Variant	Const.	Relev.	Spec.	Conc.
Yi	Full Algorithm (Ours)	**3.9**	**3.8**	**3.8**	**3.8**
	w/o Template Construction	3.6	3.5	3.4	3.5
	w/o Plan Generation	3.8	3.8	3.8	3.8
	w/o Population Initialization	3.9	3.9	3.7	3.7
	w/o Fitness Evaluation	3.7	3.7	3.7	3.7
	w/o Parent Selection & Crossover	3.9	3.8	3.7	3.7
	w/o Final Candidate Selection	3.8	3.7	3.6	3.7
Qwen	Full Algorithm (Ours)	**3.8**	**3.8**	**3.7**	**3.7**
	w/o Template Construction	3.5	3.5	3.4	3.4
	w/o Plan Generation	3.7	3.7	3.6	3.6
	w/o Population Initialization	3.8	3.8	3.6	3.6
	w/o Fitness Evaluation	3.6	3.6	3.6	3.5
	w/o Parent Selection & Crossover	3.7	3.7	3.6	3.6
	w/o Final Candidate Selection	3.6	3.6	3.5	3.5
DeepSeek	Full Algorithm (Ours)	**3.5**	**3.6**	**3.6**	**3.5**
	w/o Template Construction	3.2	3.3	3.2	3.2
	w/o Plan Generation	3.4	3.5	3.5	3.4
	w/o Population Initialization	3.5	3.5	3.4	3.4
	w/o Fitness Evaluation	3.3	3.3	3.3	3.3
	w/o Parent Selection & Crossover	3.4	3.4	3.3	3.3
	w/o Final Candidate Selection	3.3	3.3	3.2	3.3
Oss.	Full Algorithm (Ours)	**4.0**	**4.1**	**4.0**	**3.8**
	w/o Template Construction	3.7	3.8	3.7	3.6
	w/o Plan Generation	3.9	4.0	3.9	3.7
	w/o Population Initialization	4.0	4.0	3.9	3.8
	w/o Fitness Evaluation	3.8	3.9	3.8	3.7
	w/o Parent Selection & Crossover	3.9	3.9	3.8	3.7
	w/o Final Candidate Selection	3.8	3.8	3.7	3.7
Phi	Full Algorithm (Ours)	**4.3**	**4.3**	**4.2**	**4.3**
	w/o Template Construction	3.6	3.5	3.4	3.5
	w/o Plan Generation	3.8	3.8	3.8	3.8
	w/o Population Initialization	3.9	3.9	3.7	3.7
	w/o Fitness Evaluation	3.7	3.7	3.7	3.7
	w/o Parent Selection & Crossover	3.9	3.8	3.7	3.7
	w/o Final Candidate Selection	3.8	3.7	3.6	3.7

Table 29: Ablation study of various components within the genetic algorithm across the four automated metrics on **feedback generation** for all models.

Model	Variant	Const.	Relev.	Spec.	Conc.
Yi	Ours (Abstract + Summary + Strengths)	**3.9**	**3.8**	**3.8**	**3.8**
	w/o Abstract	3.6	3.7	3.5	3.6
	w/o Summary	3.7	3.7	3.6	3.7
	w/o Strengths	3.8	3.8	3.6	3.7
	w/o Abstract + Summary	3.5	3.6	3.4	3.5
	w/o Abstract + Strengths	3.6	3.7	3.5	3.6
Qwen	Ours (Abstract + Summary + Strengths)	**3.8**	**3.8**	**3.7**	**3.7**
	w/o Abstract	3.5	3.5	3.4	3.4
	w/o Summary	3.6	3.6	3.5	3.5
	w/o Strengths	3.7	3.7	3.5	3.6
	w/o Abstract + Summary	3.4	3.5	3.3	3.4
	w/o Abstract + Strengths	3.5	3.6	3.4	3.5
DeepSeek	Ours (Abstract + Summary + Strengths)	**3.5**	**3.6**	**3.6**	**3.5**
	w/o Abstract	3.2	3.3	3.2	3.2
	w/o Summary	3.3	3.4	3.3	3.3
	w/o Strengths	3.4	3.5	3.3	3.4
	w/o Abstract + Summary	3.1	3.2	3.1	3.2
	w/o Abstract + Strengths	3.2	3.3	3.2	3.2
Oss.	Ours (Abstract + Summary + Strengths)	**4.0**	**4.1**	**4.0**	**3.8**
	w/o Abstract	3.7	3.8	3.7	3.6
	w/o Summary	3.8	3.9	3.8	3.7
	w/o Strengths	3.9	4.0	3.8	3.7
	w/o Abstract + Summary	3.6	3.8	3.5	3.6
	w/o Abstract + Strengths	3.7	3.9	3.6	3.7
Phi	Ours (Abstract + Summary + Strengths)	**4.1**	**4.2**	**4.0**	**4.3**
	w/o Abstract	3.8	4.0	3.7	4.1
	w/o Summary	3.9	4.0	3.8	4.0
	w/o Strengths	4.0	4.1	3.8	4.1
	w/o Abstract + Summary	3.6	3.9	3.5	3.9
	w/o Abstract + Strengths	3.7	3.9	3.6	4.0
	w/o Summary + Strengths	3.7	3.9	3.6	4.0

Table 30: Planner ablation study across the four automated metrics on **feedback generation** for all models.

Model	Variant	Const.	Relev.	Spec.	Conc.
Yi	Full Fitness Function	**3.9**	**3.8**	**3.8**	**3.8**
	w/o Length Score (sc_{len})	3.6	3.6	3.5	3.5
	w/o Template Adherence (sc_{temp})	3.7	3.7	3.6	3.6
	w/o Readability (sc_{read})	3.8	3.8	3.7	3.7
Qwen	Full Fitness Function	**3.8**	**3.8**	**3.7**	**3.7**
	w/o Length Score (sc_{len})	3.5	3.5	3.4	3.4
	w/o Template Adherence (sc_{temp})	3.6	3.6	3.5	3.5
	w/o Readability (sc_{read})	3.7	3.7	3.6	3.6
DeepSeek	Full Fitness Function	**3.5**	**3.6**	**3.6**	**3.5**
	w/o Length Score (sc_{len})	3.2	3.3	3.2	3.2
	w/o Template Adherence (sc_{temp})	3.3	3.4	3.3	3.3
	w/o Readability (sc_{read})	3.4	3.5	3.3	3.4
Oss.	Full Fitness Function	**4.0**	**4.1**	**4.0**	**3.8**
	w/o Length Score (sc_{len})	3.7	3.6	3.5	3.7
	w/o Template Adherence (sc_{temp})	3.8	3.7	3.6	3.7
	w/o Readability (sc_{read})	3.9	4.0	3.8	3.7
Phi	Full Fitness Function	**4.3**	**4.3**	**4.2**	**4.3**
	w/o Length Score (sc_{len})	3.8	3.7	3.8	3.6
	w/o Template Adherence (sc_{temp})	3.7	3.6	3.5	3.7
	w/o Readability (sc_{read})	4.1	4.0	4.0	4.1
	w/o Forbidden Term Penalty (pen_{forb})	4.2	4.2	4.1	4.3

Table 31: Ablation study on the components of the fitness function in the **feedback generation** approach across Constructiveness (Const.), Relevance (Relev.), Specificity (Spec.), and Conciseness (Conc.) for all models.

GPUs	Wall-clock Time (Best, h)	Wall-clock Time (Worst, h)
1	2.99	5.77
4	0.75	1.44
8	0.37	0.72

Table 32: Estimated wall-clock time for processing 5,000 reviews using multi-GPU parallelization.