

Multi-Agent LLM Annotation and Scoring for Training Fine-Grained K–12 Writing-Feedback Models

Justin O. Barber and Michael P. Hemenway and Martha Bellows and Susan M. Lottridge

Pearson

{justin.barber, michael.hemenway, martha.bellows, sue.lottridge}@pearson.com

Abstract

Fine-grained formative writing feedback requires models trained on dense, standards-aligned labels. Such feedback can consist of span-level evidence markers, sentence-level error flags, and essay-level binary scores across dozens of codes. Human annotation at this granularity is the binding constraint on coverage. We describe a multi-agent large language model (LLM) pipeline that produces *verified* silver labels for this setting and then trains small, deterministic transformer scorers from them. Four agents built on Claude Sonnet 4.6 separate the work of annotation, conventions analysis, backreading, and adjudication, and three generation agents supply synthetic essays, minimal-pair counterfactuals, and construct probes that redress severe class imbalance. We evaluate on a Grade 5 opinion-writing pilot ($N=500$ essays) against a human-scored held-out set that was reserved for final evaluation. Models trained only on silver labels reach Cohen’s κ up to 0.92 on rule-amenable conventions and substantial agreement on several content codes, while subjective constructs remain a firm ceiling. The pipeline trains scorers for codes whose human labels suffice only for a small test set, at no additional human cost.

1 Introduction

Formative writing feedback must be actionable. A student and a teacher need to know not only that a response scored poorly, but which specific writing skills produced that score and where in the text the evidence lies (Shute, 2008; Graham et al., 2015). Feedback at this resolution depends on models trained against fine-grained labels. Those labels are span markers for a feature such as a topic introduction or a run-on sentence, sentence-level error flags, and essay-level binary scores across dozens of standards-aligned codes.

The binding constraint is labeled data. Human span-level annotation for more than two dozen

codes across thousands of essays is slow and onerous. For our Grade 5 pilot, scoring 500 essays on a subset of codes consumed substantial rater time. The time constraints in the pilot meant that only a subset of codes could be human scored for training; the remaining codes had fifty human scored responses but training data needed to be obtained via agentic methods.

A direct alternative, scoring each essay with a frontier LLM at inference time, is unsuitable for high-volume formative assessment. Per-response generative inference is costly and slow at the target scale. Its stochastic outputs also cannot be held to the reproducibility standard that educational measurement requires, under which the same response must receive the same score across administrations. A hosted model carries a further risk, since the provider or a regulator can change or withdraw it without notice, which breaks operational continuity.

We take a different position. The LLM operates upstream as a data-enhancement tool. A multi-agent pipeline annotates real student essays, generates additional essays to balance rare classes, and verifies every label through backreading and adjudication. The verified labels then train small open-source transformers (Longformer and DeBERTa-v3) that are deterministic, ownable, and inexpensive to serve. The generation cost is paid once and amortized across the deployment lifetime.

This study asks three questions. (RQ1) Can transformer scorers trained only on verified silver labels reach the agreement with human raters that operational assessment requires? (RQ2) Which construct types are amenable to silver-label training, and which are not? (RQ3) Does targeted generation for class balance improve agreement on the rare-class codes that motivate it?

No supervised baseline is available for these codes. Their human labels number only about fifty per code, enough for a small test set but far too few

to train a comparison model on human scores, and an LLM-at-inference system is ruled out by the cost and reproducibility constraints above. We therefore frame evaluation against human raters rather than against a competing model. Each model is judged by how closely it agrees with the human scores, and we interpret that agreement in two ways. First, we compare it against fixed benchmarks for substantial agreement (Landis and Koch, 1977; Williamson et al., 2012). Second, because the pilot collected a single human score per essay, we cannot estimate a human agreement ceiling from our own data. We instead read weak agreement on subjective constructs in light of prior work, which finds that human raters agree only modestly on such constructs, so a low score there reflects the difficulty of the construct as much as a shortcoming of the model.

Our contributions are as follows.

1. A four-agent labeling pipeline that produces span-level (BIO), sentence-level, and essay-level scores for 26 binary feedback codes, with backreading and adjudication as explicit verification stages (§4).
2. Three targeted generation procedures, persona-conditioned synthetic essays, minimal-pair counterfactuals, and construct probes, that correct class imbalance and are selected per code by hyperparameter search (§5).
3. An evaluation on a Grade 5 opinion pilot showing that silver-only training reaches κ up to 0.92 on rule-amenable codes and substantial agreement on several content codes, with an honest account of where the approach fails (§7).

2 Related Work

Automated essay scoring and feedback. Automated essay scoring has moved from feature-engineered systems (Attali and Burstein, 2006) through neural models (Taghipour and Ng, 2016) to fine-tuned transformers on rubric-aligned labels (Crossley et al., 2024; Misgna et al., 2025). The PERSUADE 2.0 corpus (Crossley et al., 2024) offers a large argumentation benchmark, though it targets secondary students and holistic scoring rather than fine-grained formative codes. Argument and discourse mining frames writing features such as claims and evidence as labeled spans (Persing and Ng, 2016; Stab and Gurevych, 2017), the same

general formulation our span-level codes use, and grammatical error annotation (Bryant et al., 2017) is a precedent for our conventions codes.

LLMs as annotators. A growing body of work uses LLMs to label text, with strong results on objective tasks and mixed results on subjective ones (Ding et al., 2023; He et al., 2024; Gilardi et al., 2023). Most of this work treats LLM output as the end product. We instead use the LLM as a labeling engine for a downstream distillation step, in the spirit of cost-reduction (Wang et al., 2021) and symbolic knowledge distillation (West et al., 2022). This separation keeps the deployed scorer deterministic and auditable while retaining the LLM’s labeling reach. It also connects to weak supervision, where noisy label sources are combined and denoised before training (Ratner et al., 2017). Our backreader and adjudicator play the denoising role through agent disagreement rather than through a generative label model.

Data generation and counterfactuals. Counterfactually augmented data improves robustness by isolating the feature that determines a label (Kaushik et al., 2019). We apply this idea to writing codes through minimal edits that flip a single trait. Persona-conditioned generation supplies controlled variety for rare classes, and construct probes test whether a model responds to the intended skill rather than a surface correlate.

Encoders and fine-tuning. We build on Longformer (Beltagy et al., 2020) for essay-length input and DeBERTa-v3 (He et al., 2021) for sentence-level classification, with discriminative learning rates (Howard and Ruder, 2018) and Optuna (Akiba et al., 2019) for search. ModernBERT (Warner et al., 2025) was a candidate backbone but did not improve over Longformer in our setting. Span labels use the BIO chunking scheme (Ramshaw and Marcus, 1995), and agreement is reported with chance-corrected statistics (Cohen, 1960, 1968; Landis and Koch, 1977) against operational evaluation guidance (Williamson et al., 2012).

3 Assessment Context and Task

The pilot targets one Grade 5 opinion-writing prompt from a formative administration. From 1,518 available responses we sampled 500, and the same set was used for every code. The NAVVY rubric decomposes five writing and language competencies, aligned to widely adopted state stan-

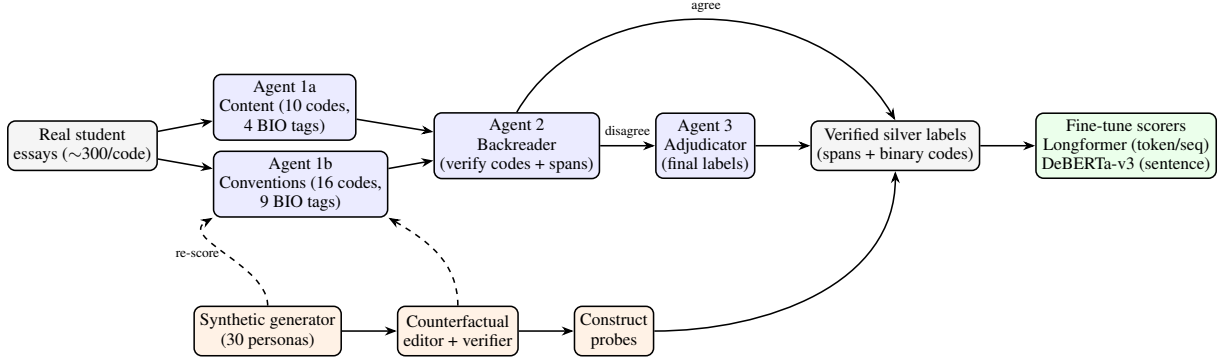


Figure 1: The labeling and generation pipeline. Two annotator agents label real essays, a backreader verifies every binary code and span, and an adjudicator resolves flagged disagreements. Synthetic and counterfactual essays re-enter the same annotation pipeline with no marker of their origin, while construct probes carry labels by design. Verified labels then train small transformer scorers.

dards, into binary component codes that deliver targeted feedback (for example, *clearly state an opinion*, *avoid run-on sentences*, and *use end punctuation*).

This paper concerns 26 component codes whose labels come from the LLM pipeline rather than from human raters. These codes admit rule-like evaluation against an explicit rubric, which makes them suited to LLM labeling. We do not treat the competency scores or the components trained on conventional human labels. Of the 26 codes, 19 had a human-scored sample available for held-out evaluation, and we report those 19 below. Each code carries an identifier of the form `<standard>-C<n>` (for example, `W51-C1` is component 1 of Grade 5 writing standard `W.5.1`); Appendix A lists every code with its short name and full rubric description.

4 Multi-Agent Annotation Pipeline

Figure 1 shows the pipeline. Every agent runs on Claude Sonnet 4.6 through AWS Bedrock and is driven by two messages. The *system* message fixes the agent’s role, a shared grade-level calibration block, the rubric definitions, and the output schema, so it encodes the standard rather than any one essay. The *user* message carries the variable content, namely the instruction, the essay, and any upstream agent outputs for the review stages. This split keeps the judgment standard stable and lets one system prompt serve the whole response set.

Annotators (Agents 1a, 1b). Two annotators divide the work by construct family. The content annotator handles ten opinion-writing and rhetoric codes and tags four span types (`TOPIC_INTRO`,

`OPINION_STATE`, `CONNECTING_WORDS`, `CONCLUSION`). The conventions annotator handles sixteen grammar and mechanics codes and tags nine span types for errors such as run-ons, fragments, and capitalization. Each produces a binary decision per code and, where a code is locatable, a verbatim span. Spans must match the essay character-for-character so they can train token-level models. Five human-labeled essays, selected by greedy coverage over the code set, seed each annotator as few-shot exemplars.

Backreader (Agent 2). The backreader reviews both annotators against the essay. It checks every binary decision, verifies span accuracy for the token-level codes, and enforces cross-code consistency rules (for example, the topic introduction and the opinion statement must occupy different spans). It assigns an agreement level and issues explicit overrides and span corrections. When it issues no override, the annotation is complete. Otherwise the case advances to adjudication. The backreader critiques existing annotations rather than producing new ones from scratch.

Adjudicator (Agent 3). When the backreader overrides at least one code, the adjudicator produces the final, authoritative codes and corrected spans from the essay and all upstream outputs. Its decisions are written to the label record.

The pipeline mirrors a human scoring process whereby every response gets a backread. Disagreement among agents, rather than a single model’s confidence, is the signal that triggers review.

5 Generation for Class Balance

Many codes are severely imbalanced in real essays. Nearly every student uses end punctuation, which leaves few negative examples for that code. Three generation procedures supply the missing cases. Every generated essay passes through the same annotation pipeline as the real essays, with no marker of its synthetic origin, so its labels are produced under identical standards. A generated essay inherits the train, validation, or test assignment of the real seed essay it was built from, so no synthetic relative of a validation or test essay can appear in training.

Synthetic essays. A generator writes new opinion essays in a Grade 5 voice from one of 30 student personas. Each persona fixes an opening style, organization, tone, error profile, and length, which keeps the set varied rather than 30 copies of one ideal essay. A vocabulary policy and an error taxonomy steer the output toward authentic developmental writing.

Counterfactuals. For codes that remain imbalanced after the synthetic-essay step, a pair of agents produces minimal pairs (Kaushik et al., 2019). An editor agent makes the smallest edit that flips one code while preserving the student’s stance, voice, and other traits. A second agent, the verifier, checks that edit and returns accept, revise, or reject. A revise routes back to the editor up to twice, and a reject drops the source. This procedure yields targeted negatives or positives without disturbing the rest of the label vector.

Construct probes. For twelve codes where one or two sentences isolate the target skill, a generator writes short snippets that each contain exactly one deliberate error or one correct demonstration. The intended label follows from the design of the snippet. Probes serve both as training data and as a diagnostic that a model responds to the construct rather than to essay-length surface features.

Per-code selection. Not every source helps every code. The three generated types are toggled per code by the hyperparameter search (§6). Across the 26 codes, 17 models used counterfactuals, 17 used synthetic essays, and 12 used construct probes, while five used real essays alone.

Composition. All 26 models share a base of 300 real student essays, drawn from a pool of 450 real

Code	Name	Real	Gen.	TrainN
W51-C1	Intro topic	300	132	432
W51-C2	States opinion	300	178	478
W51-C3	Org. structure	300	848	1148
W51-C7	Link opinion	300	84	384
W51-C8	Conclusion	300	209	509
W51-C9	Paragraphs	300	706	1006
W51-C11	Avoid repetition	300	128	428
W54-C1	Writes clearly	300	800	1100
W54-C5	On task	300	839	1139
W54-C6	Appropriate style	300	128	428
L51-C1	Conjunctions	300	134	434
L51-C2	Prepositions	300	–	300
L51-C9	Verb tense	300	–	300
L51-C11	Sentence variety	300	839	1139
L51-C12	Run-ons	300	141	441
L51-C13	Fragments	300	529	829
L51-C14	Grade level	300	800	1100
L51-C15	Proper spacing	300	880	1180
L52-C15	Commas	300	35	335
L52-C16	End punctuation	300	–	300
L52-C17	Capitalization	300	–	300
L52-C18	Quotation marks	300	–	300
L52-C19	Grade level	300	706	1006
L53-C1	Vary for interest	300	848	1148
L53-C6	Formality	300	128	428
L53-C7	Grade level	300	706	1006

Table 1: Training-data composition for the 26 silver-labeled components. All models share a base of **300 real student essays**, drawn from 450 real essays whose remainder is held out for validation and test. *Gen.* is the number of generated records added (synthetic essays, counterfactuals, and construct probes), equal to $\text{TrainN} - 300$; counterfactuals were produced from both real and synthetic essays. Five components trained on the 300 real essays alone.

essays whose remainder is held out for validation and test. Table 1 reports, per code, that real base, the number of generated records the search added on top of it, and the resulting training size. The generated share ranges from none (five codes train on the 300 real essays alone) to roughly three quarters of the set for the most imbalanced codes.

6 Training and Model Selection

Architectures. Three architectures match three construct shapes. The *token* model is Longformer-base-4096 with a BIO head. It labels each sub-word and derives an essay-level decision by counting predicted spans against a per-code threshold. The *sequence* model is Longformer-base-4096 with a binary head that scores the full essay from its pooled representation. The *sentence* model is DeBERTa-v3-base applied to individual sentences, each prefixed with its ordinal position, with a polarity-aware rule that aggregates the sentence predictions into an

essay decision. DeBERTa suits short inputs better than Longformer. The 13 span-bearing codes were trained under both the token and sequence heads, and the stronger variant was deployed. Codes without spans were trained as sequence scorers, and a subset whose skill is isolated by one or two sentences used the sentence model.

Search and selection. Hyperparameters were tuned with Optuna 3.6 using the TPE sampler (multivariate=True, group=True). The space spans modeling choices and data choices searched jointly. Modeling choices include the encoder and head learning rates searched independently (Howard and Ruder, 2018), the number of frozen lower layers, the batch size, and the sequence length. Data choices toggle the inclusion of counterfactuals, synthetic essays, and construct probes, the exclusion of ceiling-saturated records, and a switch to use only pipeline-verified labels. One metric governs the whole search, namely Cohen’s κ on the validation split. It serves as the optimization objective, the early-stopping criterion, and the checkpoint-selection criterion. Holding these three to one metric is a methodological requirement, since optimizing one metric and selecting on another would invalidate the search.

Validation and final test. Hyperparameter search, early stopping, and model selection used the silver-labeled validation split, never the human-scored data. Two further checks sit downstream. A *development held-out* split ($n=70$) carries silver labels and is withheld from both training and selection. Because its labels come from the pipeline, it measures how faithfully the model reproduces the pipeline’s labeling on unseen essays, which is a read on generalization before any human is involved. *Construct probe* pairs confirm sensitivity to the target skill. Separate from both is the human-scored gold set, which carries expert labels and is the only source that measures agreement with humans. We reserved it for a single final evaluation. For each silver-labeled code with human data available, a 50-essay gold sample was preserved for that final test and was never used for tuning or selection.

7 Results

We report on the held-out gold test set described above. Each code’s 50-essay sample received single expert scores. With $n=50$ and several low-

Code	Name	Val. (90)	Dev (70)
W51-C1	Intro topic	0.853	0.784
W51-C3	Org. structure	0.795	0.771
W51-C8	Conclusion	0.855	0.845
W54-C5	On task	0.917	0.736
L51-C14	Grade level	0.815	0.723
L52-C17	Capitalization	0.779	0.588
L52-C19	Grade level	0.658	0.705
L53-C7	Grade level	0.762	0.818
W51-C2	States opinion	0.917	0.636
L51-C11	Sentence variety	0.782	0.139
L51-C13	Fragments	1.000	0.113
L53-C6	Formality	0.685	0.470

Table 2: Silver-label κ on the validation and development held-out splits (selected codes). The top block holds stable codes. The bottom block holds large validation-to-held-out drops that flag noisy labels.

prevalence codes, κ estimates carry wide intervals. A single additional misclassification shifts κ by a median of about 0.06 across these codes (interquartile range 0.05–0.08) and by as much as 0.15–0.21 for the most imbalanced codes, so we read these values as directional rather than precise.

Silver generalization gap. Table 2 contrasts κ on the validation split with the development held-out split. Performance drops from validation to held-out, as expected, and the size of the drop is itself diagnostic. Codes with clear, locatable evidence (organization, conclusion, and grade level) hold up. Codes that depend on diffuse judgment (sentence variety and fragments) fall sharply, which signals that their silver labels carry more noise than signal.

Construct probes. Across the trained models, 9 passed both probe items, 12 passed one, and none failed both. Eleven of the partial cases missed the off-construct (negative) probe, which indicates a tendency to mark the skill present when it is absent rather than the reverse. Probes thus localize a specific failure mode that aggregate agreement obscures.

Engine–human agreement. Table 3 and Figure 2 report engine–human κ on the held-out test set for the 19 silver-trained components with a human-scored sample. Rule-amenable conventions reach the top of the range, with quotation marks at $\kappa=0.92$, end punctuation at 0.78, and capitalization at 0.71. Several content codes reach substantial agreement, including organization at 0.83, paragraphs at 0.79, and writes clearly at 0.78. The weakest results fall on subjective or diffuse codes,

Code	Name	Arch	κ	EA	F1 ₊
L52-C18	Quotation marks	Seq	0.922	0.980	0.933
W51-C3	Org. structure	Seq	0.827	0.920	0.938
W51-C9	Paragraphs	Seq	0.786	0.900	0.921
L52-C16	End punctuation	Seq	0.783	0.920	0.947
W54-C1	Writes clearly	Seq	0.781	0.900	0.923
L52-C17	Capitalization	Seq	0.711	0.860	0.881
L51-C14	Grade level	Seq	0.680	0.840	0.857
W51-C2	States opinion	Sent	0.623	0.920	0.955
L51-C11	Sentence variety	Seq	0.594	0.800	0.833
W51-C1	Intro topic	Seq	0.569	0.800	0.844
W51-C8	Conclusion	Tok	0.545	0.760	0.769
L51-C12	Run-ons	Tok	0.532	0.760	0.760
L51-C13	Fragments	Tok	0.458	0.820	0.886
L51-C9	Verb tense	Seq	0.432	0.880	0.932
W54-C5	On task	Seq	0.408	0.900	0.945
L51-C1	Conjunctions	Seq	0.393	0.740	0.822
W51-C11	Avoid repetition	Seq	0.369	0.740	0.817
W51-C7	Link opinion	Tok	0.330	0.740	0.831
L53-C6	Formality	Seq	0.168	0.582	0.568

Table 3: Engine–human agreement on the held-out test set for the 19 silver-trained components with a human-scored sample ($n=50$, except L53-C6 at $n=91$). Arch gives the architecture, where Seq is sequence, Sent is sentence, and Tok is token (BIO). EA is exact agreement and F1₊ is positive-class F1. Three further silver codes are omitted, namely L52-C15 (no positive instances in the sample) and L52-C19 and L53-C7 (no human scores available).

with formality at 0.17, link opinion at 0.33, and avoid repetition at 0.37. The quotation-marks code shows that a rule-amenable construct can be scored at near-human agreement from LLM-generated training data alone, with no human labels in its training set.

8 Discussion

Where silver labels suffice. The strongest results share a profile. The code has a clear operational definition and a discrete, observable manifestation in the text. Conventions such as quotation marks and end punctuation, and structural codes such as organization and paragraphing, clear or approach the substantial-agreement threshold. For these codes the rubric is close to a rule, the LLM applies it consistently, and the trained model recovers that consistency at a fraction of the serving cost. Back-reading and adjudication matter here, since they denoise the labels before they reach the student model, in the manner of weak supervision (Ratner et al., 2017).

Where it does not. Two failure patterns recur. First, subjective constructs set a ceiling that no label source can lift. Formality is a construct on

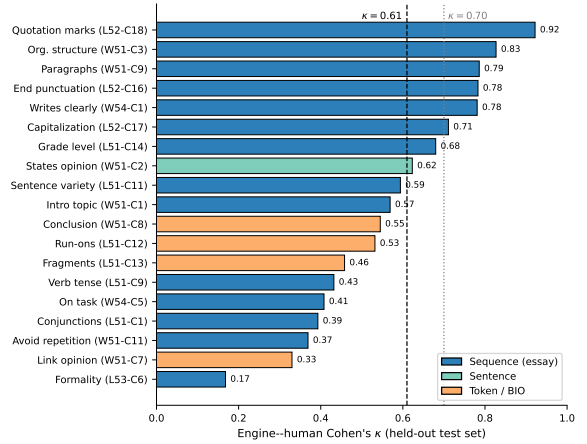


Figure 2: Engine–human κ on the held-out test set, by architecture. Reference lines mark the Landis–Koch boundary for substantial agreement ($\kappa=0.61$) and the ETS operational target ($\kappa=0.70$). With $n=50$ and imbalanced classes, the point estimates are volatile, so we read $\kappa=0.61$ as the more appropriate bar for this pilot. Rule-amenable conventions cluster at the top, and subjective constructs fall below both lines.

which human judgment is itself unsteady, and the silver model lands lower still. When the target resists consistent human scoring, neither the LLM nor the trained model can exceed the signal in the labels, and these codes need rubric refinement before they are automatable. Second, codes whose evidence sits in a narrow region of a long essay, such as a single linking phrase or a concluding sentence, present a low signal-to-noise ratio to a full-essay model. The construct probes expose this as a bias toward predicting the skill present.

9 Conclusion

A multi-agent LLM pipeline, with explicit back-reading and adjudication and with targeted generation for class balance, produces verified silver labels that train small, deterministic transformers. On a Grade 5 opinion pilot, silver-only models reach κ up to 0.92 on rule-amenable conventions and substantial agreement on several content codes, while subjective constructs remain bounded by human agreement. The pipeline extends training coverage to codes whose human labels suffice only for a small test set, at no added human cost, and points toward joint modeling across related codes as the next step.

Limitations

We read every agreement figure against its base rate and sample size, and we flag the volatility of κ at $n=50$. The validation-to-held-out drop in Table 2 is a deliberate disclosure, since validation agreement overstates field performance. The five annotator exemplars were drawn from the same 50-essay set later used for human comparison, which couples seeding to evaluation, and future runs will seed from non-evaluation data. The pipeline labels each code independently, so it does not yet exploit relationships among codes. Results are limited to one grade, one genre, and one prompt, and transfer requires fresh rubrics, calibration, and validation.

Ethics Statement

This system produces feedback that real students and teachers will see. Scores supplement teacher judgment rather than replace it, and they inform instruction rather than grades or advancement. We validate performance against human-scored benchmarks and document known limitations rather than aggregate them away, particularly on subjective and low-prevalence codes. Synthetic and counterfactual essays were used for training only and were verified through the same pipeline as authentic responses. Demographic subgroup and fairness analysis is required as the sample grows and was not available for this pilot.

References

- Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.
- Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3).
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 1: Long Papers)*, pages 793–805.
- Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.
- Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.
- Scott A Crossley, Yu Tian, Perpetual Baffour, Alex Franklin, Meg Benner, and Ulrich Boser. 2024. A large-scale corpus for assessing written argumentation: Persuade 2.0. *Assessing Writing*, 61:100865.
- Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023. Is gpt-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11173–11195.
- Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.
- Steve Graham, Michael Hebert, and Karen R Harris. 2015. Formative assessment and writing: A meta-analysis. *The elementary school journal*, 115(4):523–547.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.
- Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024. Annollm: Making large language models to be better crowdsourced annotators. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 165–190.
- Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.
- Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.
- J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.
- Haile Misgna, Byung-Won On, Ingyu Lee, and Gyu Sang Choi. 2025. A survey on deep learning-based automated essay scoring and feedback generation. *Artif. Intell. Rev.*, 58(2):36.
- Isaac Persing and Vincent Ng. 2016. End-to-end argumentation mining in student essays. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third workshop on very large corpora*.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB endowment. International conference on very large data bases*, 11(3):269.

Valerie J Shute. 2008. Focus on formative feedback. *Review of educational research*, 78(1):153–189.

Christian Stab and Iryna Gurevych. 2017. Recognizing insufficiently supported arguments in argumentative essays. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 980–990.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1882–1891.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? gpt-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, and 1 others. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625.

David M Williamson, Xiaoming Xi, and F Jay Breyer. 2012. A framework for evaluation and use of automated scoring. *Educational measurement: issues and practice*, 31(1):2–13.

A Component Code Descriptions

Table 4 gives the full rubric description for each of the 26 silver-labeled component codes, alongside the short name used in the result tables.

Code	Name	Component description
<i>W.5.1 — Opinion writing</i>		
W51-C1	Intro topic	Clearly introduce a topic or text
W51-C2	States opinion	Clearly state an opinion
W51-C3	Org. structure	Create an organizational structure
W51-C7	Link opinion	Connect opinions and reasons with words, phrases, and clauses
W51-C8	Conclusion	Provide a conclusion that relates to the presented opinion
W51-C9	Paragraphs	Format the writing sample with paragraphs
W51-C11	Avoid repetition	Avoid repetition without purpose
<i>W.5.4 — Production of writing</i>		
W54-C1	Writes clearly	Write clearly
W54-C5	On task	Address the assigned task of the writing assignment
W54-C6	Appropriate style	Use a style of writing that is appropriate to the audience
<i>L.5.1 — Grammar and usage</i>		
L51-C1	Conjunctions	Correctly use conjunctions
L51-C2	Prepositions	Correctly use prepositions
L51-C9	Verb tense	Use consistent verb tenses
L51-C11	Sentence variety	Use a variety of sentence types. Avoid overreliance on simple sentences
L51-C12	Run-ons	Avoid run-on sentences
L51-C13	Fragments	Avoid the use of fragments
L51-C14	Grade level	Write on grade level
L51-C15	Proper spacing	Use proper spacing
<i>L.5.2 — Capitalization, punctuation, spelling</i>		
L52-C15	Commas	Include and use commas when needed
L52-C16	End punctuation	Use end punctuation
L52-C17	Capitalization	Capitalize words properly. Use upper and lower case letters when typing
L52-C18	Quotation marks	Correctly use quotation marks
L52-C19	Grade level	Write on grade level
<i>L.5.3 — Knowledge of language</i>		
L53-C1	Vary for interest	Vary sentences to increase reader interest
L53-C6	Formality	Write more formally and less conversationally
L53-C7	Grade level	Write on grade level

Table 4: The 26 silver-labeled component codes, with the short names used in Tables 2–3 and their full rubric descriptions. Codes follow the NAVVY scheme <standard>-C<n>: the prefix names a Grade 5 writing (W) or language (L) standard, and C<n> is the component index within that standard. Three components share the name *Grade level* under different standards (sentence structure and vocabulary; overall conventions; and style, tone, and sophistication).