Received: 21 April 2026; **Revised:** 26 August 2026; **Accepted:** 10 September 2026; **Published:** 19 September 2026
Archs Sci. (2026) Volume 76(3), Pages 11-19, Paper ID 2026302.

Word and Insertion-Gap Error Detection in English Learner Writing: A Cross-Corpus Study of Pretrained and Syntactic Features

Xiao Yang
School of Foreign Languages, Xijing University, Xi'an 710123, Shaanxi, China
Corresponding author: Xiao Yang (e-mail: yangxiao850101@163.com).

Abstract Grammatical error detection must identify both problematic source words and locations where material is missing. This study evaluates these decisions separately and examines whether explicit dependency features improve classifiers built on frozen pretrained representations. An audited W&I+LOCNESS partition supplies 31,214 training sentences, 2,083 tuning sentences, and 2,286 internal development sentences. External evaluation uses 2,514 FCE test sentences after excluding 181 exact normalized source overlaps with the W&I partitions. Local and contextual lexical controls are compared with a frozen BART encoder followed by linear or multilayer classifiers. Dependency-relation and finite-clause-depth features are tested in parameter-matched ablations across three training seeds. The syntax-free BART multilayer model obtains mean external word and gap detection F_1 of 48.84% and 40.29%, respectively; adding both syntactic features yields 48.89% and 40.03%. The syntactic additions do not provide a consistent improvement across both detection tasks. Uncertainty is estimated by paired resampling of the 97 external writer groups, and parser-defined structure and clean-sentence false alarms are reported separately. The contribution is a reproducible, overlap-controlled comparison of word and pure-insertion-gap detection. These span-derived detection scores do not measure correction quality or establish state-of-the-art performance.

Index Terms grammatical error detection, insertion gaps, pretrained representations, dependency syntax, learner corpora, cross-corpus evaluation

I. Introduction

Grammatical error detection (GED) assigns error labels to a source text. In writing support, identifying a problematic word and identifying a missing word require different output locations: an omission may occur between two otherwise acceptable words. A word-only label sequence cannot represent every such location directly. Reporting insertion-gap detection separately therefore makes a detector's treatment of omissions visible, without requiring it to generate a correction.

Contextual representations are an established resource for GED, and their usefulness is not a new premise of this study [1]. Sequence-labeling models have likewise been studied extensively for learner writing [2]. A more focused empirical question concerns the additional value of source syntax once a pretrained encoder already supplies sentence context. Dependency relations and subordinate-clause depth are compact descriptors, but parser errors on learner language and overlap with information already present in the encoder can limit their benefit.

A credible comparison also requires a clear account of what has been held out. Sentences from the same learner can share vocabulary, proficiency, and recurring errors; exact source repetitions can create a further connection between partitions. The BEA-2019 data release provides useful learner-corpus resources [3], while the FCE corpus offers a separate source of examination writing [4]. This study audits grouping and exact source overlap, uses only W&I data for model selection, and evaluates the resulting detectors on an overlap-controlled FCE test subset.

Three research questions organize the experiment. First, how do lexical controls, linear probes of frozen BART representations, and nonlinear probes compare for word and insertion-gap detection? Second, do dependency-relation and finite-clause-depth features improve a matched nonlinear classifier on the external corpus? Third, how do outcomes differ across parser-defined structural subsets and sentences without annotated errors? The third question is descriptive: an automatically assigned structural label does not establish human-verified syntactic complexity.

The contribution is an empirical comparison of word and gap detection with audited partitions, controlled syntax ablations, and writer-level uncertainty. Established model components are used to test the value of explicit syntax under a fixed representation and training recipe.

II. Related Work

Rei and Yannakoudakis evaluated compositional sequence-labeling models for detecting errors in learner writing [2]. Bell et al. compared contextual word representations, including ELMo, BERT, and Flair, for GED [1]. These studies motivate contextual features and also limit the novelty claim here: using pretrained representations for detection is established. MultiGED-2023 extended shared-task evaluation to multilingual token-level detection [5]. The present English-only study instead emphasizes separate source-word and pure-insertion-gap outputs, controlled syntactic ablations, and transfer between two learner-corpus collections.

Transformer pretraining supplies the encoder representations used here [6], [7]. BART was developed as a denoising encoder–decoder model; this experiment uses only its frozen encoder. No decoder is trained or executed. A linear classifier provides a low-capacity probe, while a one-hidden-layer classifier tests a more flexible mapping from the same representations. The lexical systems provide inexpensive pipeline controls; because their objectives and optimization differ, their comparison with BART systems is not an isolated causal estimate of pretraining alone.

Detection-assisted generation [8] and contemporary GEC approaches [9] address related correction tasks. Their scores cannot be inserted into the present detection comparison because the output units, references, subsets, and metrics differ. Detecting a location does not establish that it can be repaired correctly or without changing meaning.

Universal Dependencies supplies a common representation for relations and clausal attachments [10]. Stanza provides the source parser used in this experiment [11]. Parser output is treated as a noisy feature rather than a gold annotation. This distinction matters twice: the features may be wrong, and the same parser also defines the descriptive structural subsets. A strong score on one such subset does not by itself establish improved linguistic reasoning.

III. Data and Detection Labels

A. Corpora and Partition Audit

The W&I+LOCNESS v2.1 release associated with BEA-2019 supplies the development environment [3], [12]. The original release contributes 34,308 training and 4,384 development sentences. W&I learner identifiers define groups; native LOCNESS documents provide groups when learner identifiers are unavailable. The released development groups are deterministically divided into tuning and internal development partitions using the supplied preparation code. All sentences belonging to a group remain together.

Exclusions are applied in disjoint priority order. The training partition loses 2,178 sentences from learners also represented in development, 495 further sentences with normalized source overlap with development, and 421 further repeated training sources. Fifteen tuning sentences that duplicate an internal development source are removed. The retained partitions contain 31,214 training, 2,083 tuning, and 2,286 internal development sentences. No observed group or exact normalized source is shared between these three partitions. The internal development subset was inspected during earlier lexical development, so its results are exploratory development evidence rather than a newly untouched test.

External evaluation uses only the released FCE v2.1 test portion [4]. Its 2,695 sentences belong to 194 essay answers from 97 writer/script identifiers; each identifier covers two answers. The M2 sentences are mapped back to the released JSON documents by exact normalized text reconstruction. Removing 181 sentences whose normalized sources occur in any retained W&I partition leaves 2,514 sentences, still covering all 97 groups and 194 documents. FCE training and development portions are not used for fitting, threshold selection, or checkpoint selection.

Normalization for overlap checking is the case-folded, whitespace-joined source-token sequence. These checks identify exact repeated sources under that convention, not paraphrases, shared prompts, cross-corpus personal identity, or pretraining contamination. The FCE results are consequently labeled as scores on an overlap-controlled subset. They are not directly comparable with published scores on the complete FCE test set. Table 1 gives the retained denominators, and Figure 1 summarizes the evaluation separation.

Table 1: Retained data and positive-label prevalence (%). Groups are learner identifiers or native documents for W&I+LOCNESS and writer/script identifiers for FCE.

Partition	Sentences	Groups	Documents	Words	Word +	Gaps	Gap +
W&I train	31,214	2325	2817	583,579	9.61	614,793	2.32
W&I tuning	2,083	163	164	41,766	8.20	43,849	2.09
W&I internal dev.	2,286	183	186	45,121	8.03	47,407	2.11
FCE external subset	2,514	97	194	41,407	12.41	43,921	2.24

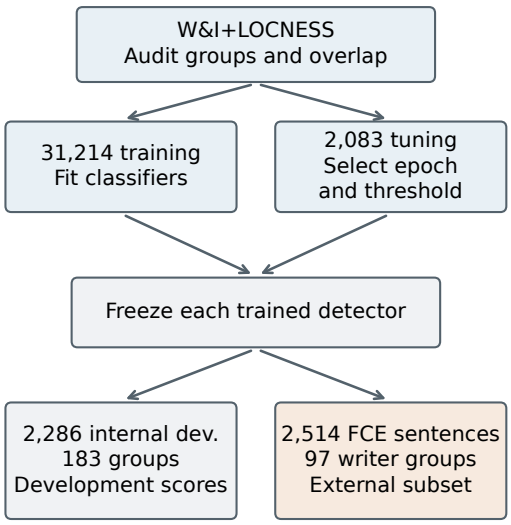

Figure 1: Executed partition and evaluation workflow. Model fitting and selection use W&I partitions only. Exact normalized source overlap is removed before external FCE scoring.

B. Source-Word and Pure-Insertion-Gap Labels

Let a tokenized source sentence be $x = (w_1, \ldots, w_n)$. It has n word decisions and $n + 1$ gap decisions, including the positions before the first word and after the last. Punctuation tokens count as words under the released M2 tokenization. Supervision comes directly from annotator 0 in the released M2 edits; references are not pooled across annotators and no new human adjudication is claimed.

An M2 edit covering a nonempty half-open source span $[a, b)$ marks every word in that span as positive. A pure insertion with $a = b$ marks gap a as positive. No-op records contribute no positive labels. Detection-only unknown-error records retain their span labels even when a replacement string is unavailable. Table 2 gives constructed examples, not quotations from the learner corpora.

$$y_i^{\mathrm{w}} = \mathbb{1}\{\exists (a, b) : a < i \leq b\}, \qquad y_j^{\mathrm{g}} = \mathbb{1}\{\exists (a, a) : a = j\}. \tag{1}$$

Here word indices are one-based and M2 offsets and gap indices are zero-based. The word rule applies only to nonempty edited spans, and the gap rule only to insertion records. This is a span-derived detection convention. A multiword replacement can mark words that a different minimal alignment would leave unchanged; an omission absorbed into a nonempty replacement is not additionally relabeled as a pure insertion. The labels therefore describe the released edit spans rather than a canonical minimal-edit alignment or all linguistically possible omissions. The included preparation script makes the convention reproducible.

Table 2: Constructed label examples. Square brackets mark positive words; ∅ marks a positive insertion gap.

Edit	Source	Reference repair
Replacement	She [go] home.	go → goes
Deletion	We discussed [about] it.	Delete about
Insertion	She is [∅] teacher.	Insert a
Initial gap	[∅] you ready?	Insert Are
No edit	They arrived yesterday.	All negative

The annotations include orthographic, punctuation, and lexical problems as well as grammatical errors. A sentence with no positive word or gap label is called *annotation-clean*, which does not certify that every possible error has been annotated. Neither a word label nor a gap label supplies a correction string.

IV. Detection Models

A. Frozen Contextual Representations

The reference encoder is `facebook/bart-base`, pinned to the revision recorded in the supplement [7]. Its six encoder layers have hidden dimension 768. The model remains in evaluation mode with all encoder parameters frozen. Only the detection heads receive gradient updates. Each released source token is mapped to its BART subwords using the tokenizer's word identifiers and the `add_prefix_space` setting. Special tokens participate in encoding but are excluded from word pooling.

For word i, the representation is the mean of its final-layer subword states, followed by normalization across its 768 coordinates:

$$\bar{h}_i = \frac{1}{|S_i|} \sum_{k \in S_i} h_k, \qquad \widetilde{h}_i = \frac{\bar{h}_i - \mu_i}{\sqrt{v_i + 10^{-5}}}. \tag{2}$$

The coordinate mean μ_i and population variance v_i are computed independently for each pooled word vector; the normalization has no learned affine parameters. Encoder features are cached once in float32 and reused across conditions and seeds. Labels and reference replacements never enter encoding or source parsing.

Sources with no words or more than 256 subwords, including special tokens, are ineligible for encoding under the fixed policy. No input is truncated. Such evaluation sentences would remain in all denominators with all-negative predictions. Parser failure alone would retain the contextual representation and assign unknown syntactic features. Actual coverage and fallback counts are reported in Section VI-A.

B. Dependency Relations and Finite-Clause Depth

Stanza 1.10.1 processes the released source tokens as pretokenized input. The English EWT POS, lemmatization, and dependency models use the recorded `ewt_nocharlm` resources and CoNLL-2017 pretrained vectors. The parser receives the erroneous source, not its correction. Token-count or token-identity mismatch, invalid heads, or dependency cycles trigger the unknown-feature fallback.

The relation feature is a one-hot vector over 46 fixed categories, including an unknown category. The explicit inventory is supplied with the code. A relation outside the inventory falls back to its base relation when available, otherwise to unknown. Clause depth is a six-category one-hot vector: 0, 1, 2, 3, 4 or more, and unknown.

A qualifying subordinate-clause head has relation `advcl`, `acl:relcl`, `ccomp`, or a `csubj` variant, and is finite. Finiteness is indicated by `VerbForm=Fin` on the head or a finite auxiliary/copula child. For each word, depth counts qualifying heads on its dependency path to the root, including itself. Depth is clipped at four. Coordination, nonfinite complements, and clauses missed by the parser need not increase this descriptor. It is a specific operational feature, not a complete linguistic measure of sentence complexity.

C. Word and Gap Classifiers

Let r_i and d_i be relation and depth one-hot vectors. A word feature vector is

$$z_i = [\widetilde{h}_i;\, m_r r_i;\, m_d d_i;\, 0], \tag{3}$$

where $m_r, m_d \in \{0, 1\}$ select the syntactic condition. A sentence boundary is represented by a zero vector with its final boundary flag set to one. Gap j receives the concatenation of its left and right word or boundary vectors:

$$z_j^{\mathrm{g}} = [z_j^{\mathrm{left}}; z_j^{\mathrm{right}}], \quad j = 0, \ldots, n. \tag{4}$$

The gap representation uses source context only; it does not contain a gold or predicted missing word. Figure 2 shows the implemented information flow.

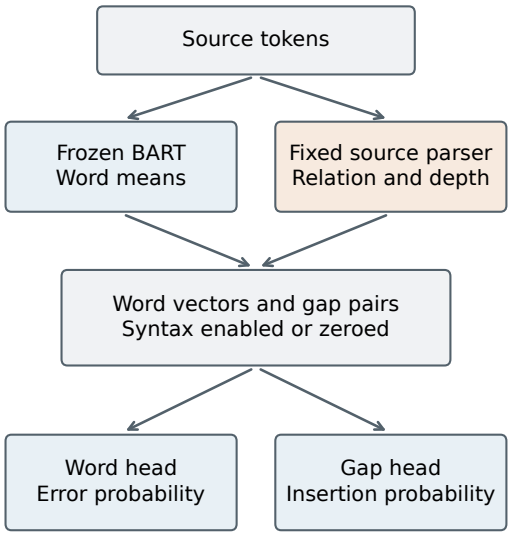

Figure 2: Implemented detector. The BART encoder and source parser are fixed. Word and gap heads are trained independently. Disabled syntactic columns are zeroed in matched ablations; no correction decoder is used.

The linear probe applies a single affine map and sigmoid with both syntax switches off. The multilayer perceptron (MLP) uses 128 hidden units, ReLU, dropout 0.2 during training, and a final affine map:

$$p(z) = \mathrm{sigmoid}(W_2\,\mathrm{Dropout}(\mathrm{ReLU}(W_1 z + b_1)) + b_2)\,. \tag{5}$$

Separate heads predict word and gap probabilities. The four MLP conditions are no syntax, relation only, depth only, and both. Input dimensions and trainable parameter counts remain identical across these four conditions; disabled columns are zeroed instead of removed. Within a seed, initial parameters, decision ordering, epochs, and optimizer settings also match. This controls nominal model capacity while changing the supplied syntactic information.

D. Lexical Controls

Two lexical feature sets are trained with independent word and gap logistic classifiers implemented in scikit-learn [13]. Word-local features include case-folded identity, case shape, digit status, capped character length, and prefixes and suffixes of lengths one to three. Word-context features additionally include the identities of two neighboring words on each side and adjacent bigrams. Gap-local features include the adjacent left and right words and a boundary indicator. Gap-context features add their joint pair and one outer word on each side.

Features use signed hashing into 2^{18} dimensions followed by row-wise L_2 normalization. Averaged stochastic gradient descent fits logistic loss for ten epochs with L_2 regularization $\alpha = 10^{-5}$, no class weighting, and no tolerance-based early stopping. These classifiers use equal decision weights, whereas the BART heads use sentence normalization. Their comparison therefore concerns complete modeling recipes. All-positive and all-negative decision rules provide prevalence-sensitive reference points without training.

V. Experimental Design

A. Fitting and Model Selection

The execution specification was recorded before fitting the new BART detection heads and before their external results were inspected. It is supplied with its SHA-256 digest; it was not publicly preregistered. The study is exploratory, including its designated main comparison. The same seeds, 13, 37, and 71, are used for every trained condition. Five BART conditions and two lexical conditions, each with word and gap heads, produce 42 trained runs in total.

Each BART head trains for ten complete epochs on eligible training decisions with AdamW, learning rate 10^{-3}, weight decay 10^{-2}, gradient norm clipping at 1, and batches of 8,192 decisions. No encoder fine-tuning, synthetic error generation, FCE adaptation, or early stopping is performed. For a task with N sentences and n_s decisions in sentence s, the objective is

$$\mathcal{L} = \frac{1}{N}\sum_{s=1}^{N}\frac{1}{n_s}\sum_{j=1}^{n_s}\mathrm{BCE}(p_{sj}, y_{sj}). \tag{6}$$

Uniformly shuffled decision batches use weights $D/(Nn_s)$, where $D = \sum_s n_s$, so each sentence has equal total loss weight. A final partial batch is retained. Word and gap objectives are optimized independently.

After each epoch, tuning scores select a threshold from $\{0.01, 0.02, \ldots, 0.99\}$ by maximum micro F_1; threshold ties favor the larger value. The checkpoint with highest tuning F_1 is retained, with earlier epochs preferred on ties. All ten epochs still run. Lexical thresholds use the same grid and tie rule after their fixed ten-epoch fit. External and internal-development scores never select a checkpoint, threshold, or hyperparameter. Table 3 records the principal settings.

Table 3: Fixed BART-head training and inference settings.

Setting	Value
Frozen encoder	BART-base; frozen
Source limit	256 subwords; no truncation
MLP hidden layer	128 ReLU units; dropout 0.2
Word / gap input	821 / 1642 dimensions
MLP parameters (word / gap)	105,345 / 210,433
Linear parameters (word / gap)	822 / 1,643
Optimizer	AdamW; learning rate 0.001
Weight decay / clipping	0.01 / gradient norm 1
Epochs / decision batch	10 / 8,192
Seeds	13, 37, 71
Thresholds	0.01–0.99; tuning micro F_1

B. Metrics and Uncertainty

Word and gap decisions are scored separately against the span-derived labels. With pooled true positives TP, false positives FP, and false negatives FN,

$$\begin{cases} P = \dfrac{TP}{TP + FP}, \\ R = \dfrac{TP}{TP + FN}, \\ F_\beta = \dfrac{(1+\beta^2)TP}{(1+\beta^2)TP + FP + \beta^2 FN}. \end{cases} \tag{7}$$

Zero-denominator scores are defined as zero. The primary metric is micro F_1. Precision, recall, and detection $F_{0.5}$ are also reported; the latter is not a correction shared-task score. A correct negative does not add to F_1, which avoids allowing the abundant negative decisions to dominate a headline accuracy statistic.

For each condition, the mean and sample standard deviation across the three training seeds describe fitting variability. For each seed, 10,000 paired bootstrap replicates resample writer groups with replacement, using random seed 1337. Confusion counts are summed over the sampled groups before recomputing F_1 and system differences. The 2.5th and 97.5th percentiles form a 95% interval. FCE uses 97 writer/script groups; internal development uses 183 learner or native-document groups. Identical resamples are used for the two systems in each comparison, following the need to match inference to the evaluation design [14].

The principal contrast is MLP with both syntactic features minus MLP without syntax on the FCE subset, separately for word and gap detection. Relation-only, depth-only, linear-probe, lexical, and subgroup comparisons are secondary. Intervals are per-seed, unadjusted exploratory intervals; they are not family-wise significance tests. Seed variability and evaluation-sample uncertainty are not combined into a single interval. These analyses describe the observed corpora and do not estimate uncertainty from choosing a different pretrained backbone.

C. Structure and False Alarms

Structural subsets are fixed from source parsing: at least one qualifying finite subordinate clause, no such detected clause, and parser unknown. Length subsets contain 1–15, 16–30, or at least 31 source tokens. All decisions in a sentence inherit its sentence-level subset. A score in the subordinate-clause subset therefore covers the entire sentence, not exclusively words inside subordinate clauses or dependencies crossing clause boundaries.

For annotation-clean sentences, false-alarm rates are the proportions receiving at least one positive word decision, at least one positive gap decision, or either type of decision. This sentence-level diagnostic answers whether the detector would flag text with no recorded edit. It does not measure whether a generated correction is necessary, useful, or meaning-preserving. Structural and false-alarm analyses reuse the already selected thresholds; no subset receives its own threshold.

VI. Results

A. Execution and Coverage

All 38,097 retained sentences were encoded without truncation, and the parser preserved every released source-token sequence. There were no overlength, alignment, or parser fallbacks in this execution. The parser detected at least one qualifying finite subordinate clause in 13,095 training, 892 tuning, 1,022 internal-development, and 995 external sentences. These coverage checks establish execution completeness, not the linguistic accuracy of the parses.

All 30 BART heads completed ten epochs, and all 12 lexical heads completed their fixed training procedure. Source feature extraction took 11.44 minutes in the completed execution, including 4.41 minutes of encoder calls and 6.71 minutes of parser calls. The BART-head ledger records 13.99 minutes of fitting, tuning evaluation, and checkpoint saving in total. These elapsed measurements exclude resource downloads and external scoring and are not isolated inference-latency benchmarks; shared CPU load can affect elapsed time. Execution used CPU only, four PyTorch threads, and one OpenBLAS thread on a machine with an eight-core quota and 20 GiB memory.

Nine constructed-data contract checks pass for gap boundaries, syntactic masking, matched MLP initialization and parameter counts, sentence weighting, clause-depth inheritance, and cycle rejection. Independently, the result audit verifies all 84 saved evaluation vectors against their reported confusion counts and metrics. Constructed checks are not counted as learner-corpus observations.

B. Overall Detection Performance

Table 4 reports all seven trained modeling conditions on internal development and external FCE. Figure 3 shows the external comparison. All means refer to separately evaluated seeds, not an ensemble. Full confusion counts, thresholds, per-seed scores, and bootstrap intervals accompany the code.

On the external subset, the syntax-free BART MLP obtains word $F_1 = 48.84\%$ and gap $F_1 = 40.29\%$. The contextual lexical controls obtain 27.87% and 11.45%, respectively. The corresponding pipeline differences are 20.98 and 28.84 percentage points. This comparison supports the usefulness of the complete pretrained-feature recipe within the study, while changing both representation and fitting procedure.

The syntax-free MLP also exceeds the linear BART probe by 3.23 word and 3.64 gap percentage points on the external subset. Because both probes share frozen encoder features, this comparison concerns the classifier mapping and its capacity. The all-positive external rules obtain 22.09% word and 4.38% gap F_1; all-negative rules have zero recall and zero F_1.

For the syntax-free MLP, internal-development scores are 50.50% for words and 52.02% for gaps. The external gap score is 11.73 points lower than the internal score, compared with 1.66 points for words. Corpus differences in language,

ARCHIVES DES SCIENCES
A Multidisciplinary Journal

Table 4: Detection performance (%). F_1 entries show three-seed mean ± sample standard deviation; precision and recall are three-seed means.

Model	Dev. word F_1	Dev. gap F_1	FCE word P	R	F_1	FCE gap P	R	F_1
Lexical local	19.54±0.05	24.97±0.09	21.48	28.40	24.46±0.08	14.61	9.96	11.84±0.06
Lexical context	22.23±0.03	25.40±0.17	20.88	41.89	27.87±0.03	13.29	10.06	11.45±0.00
BART linear	45.65±0.21	48.59±0.43	55.07	38.94	45.61±0.36	38.04	35.37	36.65±0.20
BART MLP	50.50±0.39	52.02±0.05	57.81	42.43	48.84±0.51	37.88	43.06	40.29±1.21
MLP + relation	50.92±0.12	52.50±0.04	58.13	42.30	48.95±0.29	37.41	42.95	39.96±0.54
MLP + depth	50.63±0.60	52.13±1.11	58.96	41.63	48.78±0.60	37.59	42.85	39.94±0.40
MLP + both	50.64±0.52	52.60±0.35	58.15	42.28	48.89±0.78	37.93	42.38	40.03±0.53

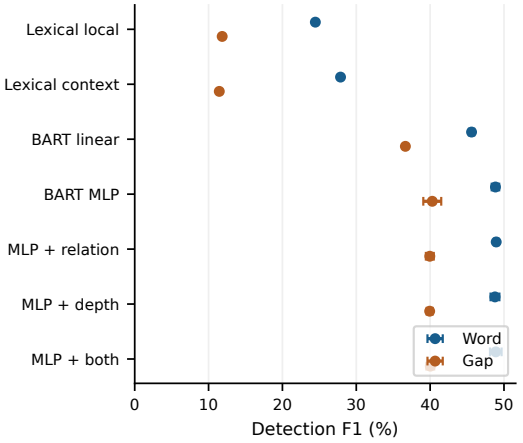

Figure 3: External FCE subset detection F_1. Points show three-seed means and bars show one sample standard deviation across seeds. These bars are not writer-bootstrap confidence intervals.

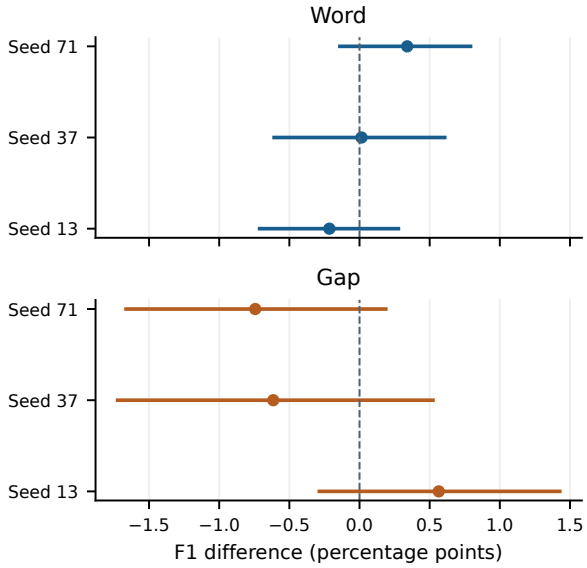

Figure 4: Both syntactic features minus no syntax on the external FCE subset. Differences are percentage points; bars are per-seed paired 95% writer-bootstrap intervals. Zero indicates no observed difference.

prevalence, and edit annotation prevent attributing this change to a single cause; it nevertheless cautions against treating internal development scores as transfer performance.

C. Syntactic Ablations

Adding relation and depth features together changes mean external F_1 by +0.05 percentage points for words and -0.26 points for gaps. Table 5 and Figure 4 show the paired uncertainty for each seed. All six intervals include zero; the data do not resolve an improvement for either task. These are unadjusted per-seed intervals, so they are interpreted as exploratory evidence rather than a family-wise declaration of significance.

The relation-only and depth-only external word means are 48.95% and 48.78%; their gap means are 39.96% and 39.94%. Reporting all four MLP conditions avoids presenting only the most favorable ablation. The combined condition is the specified comparison, not a model selected because it won on FCE. The observed syntactic effects should be assessed against both fitting variability and paired writer-level uncertainty.

D. Structural Subsets and Annotation-Clean Sentences

The external parser-defined subordinate-clause subset contains 995 sentences, compared with 1,519 without a detected qualifying clause. For the syntax-free MLP, word F_1 is 49.01% in the former subset and 48.64% in the latter; gap F_1 is 39.03% and 41.88%, respectively. Table 6 also reports label prevalence and length subsets. These descriptive differences are not controlled estimates of a complexity effect, and they do not isolate errors inside subordinate clauses.

Among 709 annotation-clean external sentences, the syntax-free MLP flags 17.21% through its word head, 9.21% through its gap head, and 23.04% through either head, averaged across seeds. The combined-syntax model flags 22.33% through either head. Thus, the F_1-selected operating points still produce substantial sentence-level false alarms under the released annotations. Higher detection F_1 should not be interpreted as evidence that feedback is sufficiently conservative for educational deployment.

Table 5: Primary external contrast: both syntactic features minus no syntax. Scores are percentages; differences and paired 95% intervals are percentage points.

Task	Seed	No syntax F_1	Both F_1	Difference	Paired 95 percent interval
Word	13	48.42	48.20	-0.21	[-0.71, +0.28]
Word	37	48.71	48.73	+0.01	[-0.61, +0.61]
Word	71	49.41	49.74	+0.34	[-0.14, +0.79]
Gap	13	38.95	39.51	+0.56	[-0.29, +1.43]
Gap	37	40.61	40.00	-0.61	[-1.72, +0.52]
Gap	71	41.31	40.56	-0.74	[-1.66, +0.19]

Each interval uses 10,000 paired bootstrap samples of 97 writer groups. Intervals are exploratory and unadjusted for multiple comparisons.

Table 6: External structural and length subsets. Prevalence and F_1 are percentages; F_1 is the mean across seeds. The structural and length partitions overlap.

Sentence subset	n	Word +	Gap +	Word no syntax	Word both	Gap no syntax	Gap both
Subordinate clause detected	995	12.92	2.31	49.01	49.15	39.03	39.06
No subordinate clause detected	1519	11.85	2.16	48.64	48.57	41.88	41.24
1–15 source tokens	1340	11.23	2.09	46.16	46.20	43.91	42.44
16–30 source tokens	990	12.57	2.13	48.34	48.34	38.56	38.76
31 or more source tokens	184	14.10	2.87	53.44	53.65	39.50	39.71

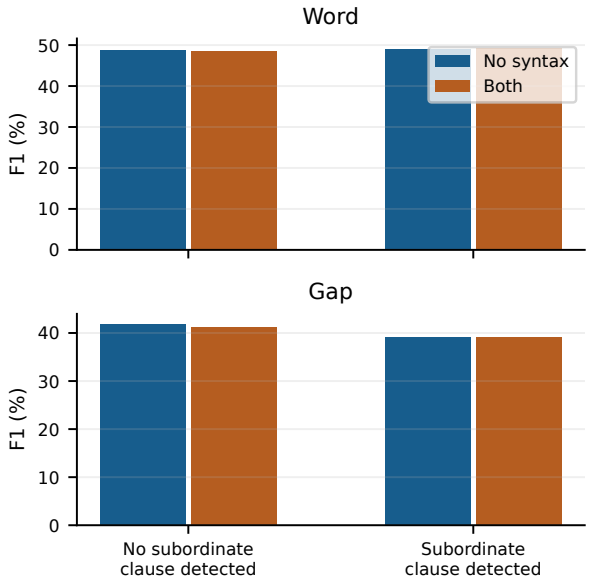

Figure 5: External FCE subset performance by parser-defined sentence structure. Values are means across three seeds. Subsets include every decision in the selected sentences and are not human-adjudicated structural categories.

E. Training Behavior and Reproducibility

The selected BART-head checkpoints range from epoch 6 to epoch 10; 14 of 30 runs select the final epoch. Figure 6 reports every epoch for the two principal MLP conditions rather than only the selected maximum. Selection at the final epoch in several runs means the fixed budget should not be interpreted as proof of convergence. The comparisons estimate performance under the specified budget; longer training could change both absolute scores and relative effects.

The archived execution specification has SHA-256 prefix `f7deb1624094e259`, and the full digest is included with the logs. Stored heads can be reloaded, and the supplied analysis recomputes metrics from their numeric prediction records. Resource and input hashes identify the evaluated data and pretrained files without distributing learner text.

Table 7: False alarms on 709 annotation-clean external sentences (%; three-seed mean ± sample standard deviation). A sentence is flagged if at least one decision is positive.

Model	Word alarm	Gap alarm	Either alarm
Lexical local	66.90±0.35	13.07±0.08	70.24±0.28
Lexical context	72.50±0.00	14.76±0.08	74.19±0.14
BART linear	13.35±0.59	7.95±0.65	19.46±0.98
BART MLP	17.21±3.30	9.21±0.45	23.04±2.65
MLP + relation	17.87±2.00	9.21±1.14	23.37±2.03
MLP + depth	16.03±2.37	9.40±1.47	22.05±1.93
MLP + both	16.55±2.62	9.12±0.50	22.33±1.63

VII. Discussion and Limitations

The pretrained-feature pipeline improves on the lexical controls, and nonlinear heads improve on linear probes. By comparison, the matched syntax additions change external word and gap F_1 by only +0.05 and -0.26 percentage points. All six paired intervals include zero. These results do not demonstrate either a consistent syntax benefit or statistical equivalence; they delimit the observed effect under this configuration.

The gap task exposes a limitation of treating all error detection as source-token classification. Pure insertions occupy positions that are absent from a word-only output sequence. However, separate gap labels do not resolve annotation ambiguity: a reference may encode an omission inside a larger replacement. The reported gap recall applies to released pure-insertion records, not to every missing expression a reader might identify. Published GED or GEC numbers obtained from different alignments or complete benchmark partitions

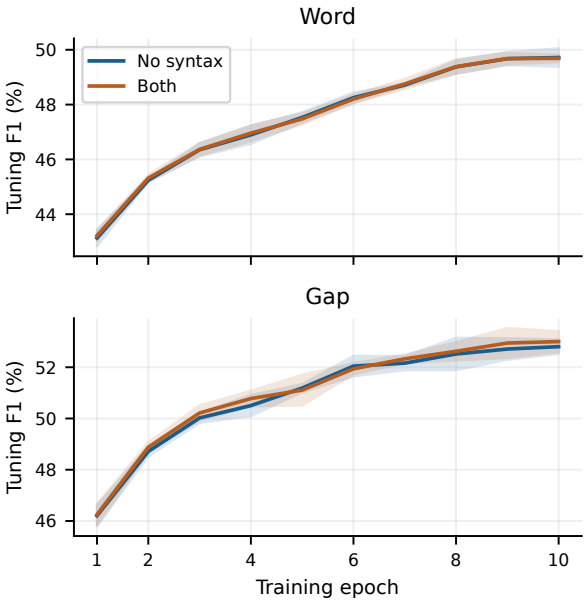

Figure 6: Tuning F_1 over all ten training epochs for the syntax-free and combined-syntax MLPs. Lines are three-seed means; shaded bands show one sample standard deviation. Each run selects its own checkpoint using tuning results only.

should not be compared directly with these tables.

Several design choices bound the findings. BART is frozen, so the study does not determine the performance of end-to-end fine-tuning or stronger contemporary detectors. The lexical comparison changes features, objective weighting, and optimization together. The matched MLP ablations provide a more controlled test of the selected syntax descriptors, but only for this representation, head family, threshold objective, and training budget. Three seeds describe limited fitting variability rather than the full space of optimization outcomes.

The parser is trained on general English resources and can misanalyze erroneous learner language. No human validation of its clause assignments is claimed. The structural subsets also differ in sentence length, error prevalence, and potentially proficiency; their raw score differences cannot isolate a causal effect of complexity. Source overlap checks remove exact normalized matches, while near duplicates, prompt overlap, unobserved cross-corpus identity, and possible exposure during pretraining remain unmeasured. External transfer between these two English learner corpora does not establish multilingual or unrestricted-domain generalization.

One annotator's spans can leave errors unrecorded and can mark partly acceptable material inside broad replacements. The resulting false alarms and misses remain reference-dependent. The micro F_1 threshold objective may also favor recall more than a writing-feedback application would tolerate. No deployment policy, learning gain, feedback-comprehension benefit, or correction-quality improvement is established by this study.

VIII. Conclusion

This study evaluates source-word and pure-insertion-gap error detection using audited W&I+LOCNESS partitions and an overlap-controlled FCE test subset. Frozen BART representations with trained MLP heads obtain mean external word and gap F_1 of 48.84% and 40.29%, exceeding the lexical and linear-probe controls in this experiment. Adding dependency relations and finite-clause depth changes these scores to 48.89% and 40.03%. The paired comparisons, structural breakdowns, and annotation-clean false-alarm rates delimit the interpretation of those scores. The resulting contribution is a reproducible cross-corpus detection study, with explicit insertion locations and controlled syntax ablations, rather than evidence about correction generation or learning outcomes.

Competing interests

The author declares no competing interests.

Funding

There is no specific funding to support this research.

Data and Code Availability

The data is available from the author on reasonable request.

Use of Generative AI

Grammar check and language edits were made using ChatGPT.

References

[1] Bell, S., Yannakoudakis, H., & Rei, M. (2019). Context is key: Grammatical error detection with contextual word representations. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 103–115). Association for Computational Linguistics.

[2] Rei, M., & Yannakoudakis, H. (2016). Compositional sequence labeling models for error detection in learner writing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1181–1191). Association for Computational Linguistics.

[3] Bryant, C., Felice, M., Andersen, Ø. E., & Briscoe, T. (2019). The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 52–75). Association for Computational Linguistics.

[4] Yannakoudakis, H., Briscoe, T., & Medlock, B. (2011). A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 180–189). Association for Computational Linguistics.

[5] Volodina, E., Bryant, C., Caines, A., De Clercq, O., Frey, J.-C., Ershova, E., Rosen, A., & Vinogradova, O. (2023). MultiGED-2023 shared task at NLP4CALL: Multilingual grammatical error detection. In *Proceedings of the 12th Workshop on NLP for Computer Assisted Language Learning* (pp. 1–16). LiU Electronic Press.

[6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 5998–6008).

[7] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7871–7880). Association for Computational Linguistics.

[8] Li, W., & Wang, H. (2024). Detection-correction structure via general language model for grammatical error correction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1748–1763). Association for Computational Linguistics.

[9] Omelianchuk, K., Liubonko, A., Skurzhanskyi, O., Chernodub, A., Korniienko, O., & Samokhin, I. (2024). Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)* (pp. 17–33). Association for Computational Linguistics.

[10] Nivre, J., de Marneffe, M.-C., Ginter, F., Hajič, J., Manning, C. D., Pyysalo, S., Schuster, S., Tyers, F., & Zeman, D. (2020). Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 4034–4043). European Language Resources Association.

[11] Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 101–108). Association for Computational Linguistics.

[12] Yannakoudakis, H., Andersen, Ø. E., Geranpayeh, A., Briscoe, T., & Nicholls, D. (2018). Developing an automated writing placement system for ESL learners. *Applied Measurement in Education, 31*(3), 251–267.

[13] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

[14] Dror, R., Baumer, G., Shlomov, S., & Reichart, R. (2018). The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1383–1392). Association for Computational Linguistics.

• • •