Vol. 1, No. 1 (2026) | pp. 55-67 | DOI: 10.5281/zenodo.23076774

Research Article

Parse-Grounded Detection of Subject–Verb Agreement Violations: Agreement-Neutralised Re-parsing versus Language-Model Scoring on BLiMP

Prakhar Gautam[1*], Dr. Jitendra Singh Thakur[2], Dr. Ashish Mishra[3]

[1] *Research Scholar, Jabalpur Engineering College, Jabalpur, Madhya Pradesh, India*
Email: prakhargautam@jecjabalpur.ac.in
[2] *Jabalpur Engineering College, Jabalpur, Madhya Pradesh, India*
Email: jsthakur@jecjabalpur.ac.in
[3] *Gyan Ganga Institute of Technology & Sciences (GGITS), Jabalpur, Madhya Pradesh, India*
Email: ashishmishra@ggits.org

Received: 12 July 2026 | Revised: 20 August 2026 | Accepted: 18 September 2026 | Published: 26 September 2026
DOI: 10.5281/zenodo.23076774

How to cite this article: Prakhar Gautam, Jitendra Singh Thakur, Ashish Mishra (2026). Parse-Grounded Detection of Subject–Verb Agreement Violations: Agreement-Neutralised Re-parsing versus Language-Model Scoring on BLiMP. *International Journal of Advanced Computing and Machine Intelligence (IJACMI)*, 1(1), pp. 55-67. DOI: 10.5281/zenodo.23076774

Abstract

Subject–verb agreement (SVA) errors are frequent in learner and machine-generated English, and minimal-pair benchmarks show that language models (LMs) still confuse the agreement controller with an intervening noun. We test whether an explicit, parse-grounded checker is a stronger detector than LM scoring. The proposed Parse-grounded Agreement Consistency Checker (PACC) locates the subject of every finite verb in a dependency parse, reads the verb's number from its surface form through a morphological lexicon, and flags conflicts. Because off-the-shelf parsers are trained on well-formed text and often misread the very verb that carries the error, PACC adds agreement-neutralised re-parsing (ANR): when the main clause has no analysable finite verb, candidate verbs are replaced one at a time by their number-neutral past-tense form, the sentence is re-parsed, and the original form is checked against the recovered subject. We evaluate on the six English SVA paradigms of BLiMP (6,000 minimal pairs), split once into 1,200 development pairs used for rule design and 4,800 held-out test pairs scored in a single run. On the test pairs, PACC with the medium spaCy parser selected on development data reaches 96.5% pairwise accuracy (95% bootstrap CI 96.0–96.9), against 86.6% (85.7–87.6) for the released GPT-2 scores of the BLiMP authors and 52.0% for a Kneser–Ney 5-gram trained here on 6.0 million tokens. The largest gain is on attractors inside relative clauses (89.4% vs 65.9% for GPT-2). A post-hoc run with a transformer-based parser raises accuracy to 98.1% and relative-clause accuracy to 98.0%, at 51 rather than 670 sentences per second. Used as a stand-alone detector on 9,600 single sentences, PACC reaches F1 = 96.3% at a 2.96% false-positive rate, while per-word LM scores barely separate acceptable from unacceptable sentences (ROC-AUC 0.578 for GPT-2). Ablations show that reading number from the surface form and ANR contribute most. Manual inspection of the remaining errors attributes them to attractor attachment, gaps in irregular-plural morphology, and at least 17 test pairs whose "acceptable" member itself contains an agreement error.

Keywords: *subject–verb agreement; grammatical error detection; dependency parsing; minimal pairs; BLiMP; language-model evaluation; agreement attraction; neuro-symbolic NLP*

1.0 Introduction

English subject–verb agreement requires a finite verb to match the number (and, for a few forms, the person) of its grammatical subject. The dependency is hierarchical rather than linear: in The painting of those vases was disturbing Rachelle, the verb agrees with painting, not with the adjacent plural vases. Humans occasionally err in exactly this configuration, a phenomenon known as agreement attraction [27], and the same configuration is where neural language models are least reliable [2], [3], [4].

Two lines of work meet at this problem. Targeted syntactic evaluation asks whether an LM assigns higher probability to the grammatical member of a minimal pair [1], [3], [5]. Grammatical error correction (GEC), in contrast, needs a system that decides for a single sentence whether it contains an error and where [15]. Modern GEC systems are sequence taggers or sequence-to-sequence models trained on large parallel corpora [16], [17], or prompted large language models [24]. Their decisions are hard to inspect, and pairwise LM preferences do not by themselves give a usable single-sentence detector.

A symbolic alternative is old but has rarely been tested against modern LMs on controlled data: parse the sentence, find each verb's subject, and compare their number. Its obvious weakness is that statistical parsers are trained on well-formed text and degrade on ungrammatical input [18]. In agreement errors this degradation is not

© 2026 The Author(s). Published by WhiteField Academic Press

random: the erroneous verb form (Craig work, The teenager like) is precisely the token a tagger is most likely to misread as a noun or preposition, so the parse that should expose the error hides it.

This paper makes four contributions:

(i) A parse-grounded agreement checker (PACC) that reads agreement from surface morphology rather than from tagger output, and resolves quantity heads, relative pronouns, expletives and coordination.

(ii) Agreement-neutralised re-parsing (ANR), a simple repair for parser fragility on ungrammatical input: candidate verbs are replaced by number-neutral past-tense forms, the sentence is re-parsed, and the original form is checked against the recovered structure.

(iii) A controlled, leakage-free comparison on the six English SVA paradigms of BLiMP [1], with a fixed development/test split, a single test run, paired significance tests, and like-for-like comparison against the per-item scores that the BLiMP authors released for an n-gram model, an LSTM, Transformer-XL and GPT-2.

(iv) An error analysis that separates parser attachment errors, lexicon gaps and label noise in the benchmark itself.

Section 2 reviews related work and Section 3 states the gap and objectives. Section 4 describes data, baselines and the proposed method, and Section 5 the experimental setup. Section 6 reports results, ablations and error analysis; Sections 7 and 8 discuss implications, limitations and future work.

2.0 Related Work

2.1 Agreement as a probe of neural language models

Linzen et al. [2] introduced number prediction on naturally occurring sentences and showed that LSTMs trained with an LM objective fail most often when nouns of the opposite number (attractors) intervene between subject and verb. Gulordava et al. [4] repeated the test on nonsense sentences to remove lexical cues and found that LSTMs still track hierarchical structure to a useful degree. Marvin and Linzen [3] built templated minimal pairs covering agreement across prepositional phrases and relative clauses, and found relative-clause attractors among the hardest cases. Mueller et al. [6] extended templated agreement tests to several languages, and Newman et al. [7] argued that minimal-pair scores should be refined so that they reflect behaviour on the verbs a model is actually likely to produce. Finlayson et al. [8] used causal mediation analysis to locate the network components that carry subject number, and Wei et al. [9] showed that a transformer's agreement accuracy on a verb depends on how often that verb occurred in pre-training.

2.2 Minimal-pair benchmarks

BLiMP [1] contains 67 automatically generated paradigms of 1,000 minimal pairs each, with 96.4% aggregate human agreement; six paradigms target English SVA. Its authors scored an n-gram model, an LSTM, Transformer-XL [20] and GPT-2 [19], and released the per-item log-probabilities. Hu et al. [5] assessed syntactic generalisation across model architectures and training-data sizes with a suite of targeted tests. MultiBLiMP 1.0 [13] extends agreement minimal pairs to over one hundred languages. Several studies question what minimal-pair accuracy measures: Vamvas and Sennrich [10] show that contrastive scores need not reflect what a model generates, Hu and Levy [11] show that prompting an LLM for a judgement underestimates what its probabilities encode, and Hu et al. [12] report that LLM probability judgements align closely with human judgements on key constructions. The CoLA corpus [14] frames acceptability as single-sentence classification, which is closer to the detection setting of GEC.

2.3 Grammatical error detection and correction

Bryant et al. [15] survey the datasets, methods and evaluation practice of GEC, in which subject–verb agreement is one of the standard annotated error types. Tagging systems such as GECToR [16] predict token-level edits, Rothe et al. [17] obtained strong multilingual results with large sequence-to-sequence models, and Loem et al. [24] studied prompted GPT-3 for correction. These systems learn agreement implicitly from parallel data and expose no explicit reason for a decision.

2.4 Parsing ungrammatical text

© 2026 The Author(s). Published by WhiteField Academic Press

Rule-based agreement checkers depend on a parser, and parsers are trained on edited, well-formed treebanks. Hashemi and Hwa [18] measured how much dependency-parser accuracy degrades on sentences containing learner and machine-translation errors. We are not aware of work that repairs this failure specifically for agreement by re-parsing a number-neutralised variant of the sentence, or that compares such a checker directly with LM minimal-pair scores on the same items.

3.0 Research Gap and Objectives

3.1 Research gap

Three gaps motivate this study. First, agreement minimal-pair work evaluates LMs almost exclusively; explicit structural checkers are rarely scored on the same items, so it is unknown how far a parse-grounded method stands above or below LM scoring on controlled agreement data. Second, the known fragility of parsers on ungrammatical input [18] has not been addressed with a repair aimed at the error type itself. Third, pairwise preference is an indirect measure for error detection: a GEC component sees one sentence, not a pair, and the single-sentence reliability of LM scores on agreement is rarely reported.

3.2 Objectives

O1. Design an interpretable, parse-grounded SVA checker that works on a single sentence and names the verb and subject in conflict.

O2. Mitigate parser failure on ill-formed verbs through agreement-neutralised re-parsing, and measure its contribution by ablation.

O3. Compare the checker with n-gram and neural LM scoring on identical held-out BLiMP items, with confidence intervals and paired significance tests.

O4. Measure single-sentence detection quality (precision, recall, F1, false-positive rate) and contrast it with the ranking quality of LM scores (ROC-AUC).

O5. Characterise the residual errors by cause, including errors in the benchmark labels.

4.0 Proposed Methodology

4.1 Task formulation

Let a minimal pair be (s^+, s^-), where s^+ is labelled acceptable and s^- differs from it only in agreement. A detector maps a sentence to a violation count $v(s) \geq 0$. It is used in two ways. In pairwise evaluation, the detector prefers the sentence with fewer violations; an LM prefers the sentence with the higher total log-probability. In single-sentence detection, a sentence is flagged as unacceptable when $v(s) > 0$, with no access to its partner.

4.2 Dataset

We use the six SVA paradigms of BLiMP [1], obtained from the official repository (github.com/alexwarstadt/blimp, directory data/, CC-BY licence). Each paradigm has 1,000 generated minimal pairs.

Table 1. The six BLiMP subject–verb agreement paradigms and the fixed development/test split (examples are development items).

Paradigm (BLiMP UID)	Short name	Contrast	Example (acceptable / unacceptable)	Dev	Test
regular_plural_subject_verb_agreement_1	Reg-1	verb form varies	Craig works with Kristin. / Craig work with Kristin.	200	800
regular_plural_subject_verb_agreement_2	Reg-2	subject number varies	The boys have kissed. / The boy have kissed.	200	800
irregular_plural_subject_verb_agreement_1	Irr-1	verb form varies, irregular nouns	An ox has looked like some print. / An ox have looked like some print.	200	800
irregular_plural_subj	Irr-2	subject number	The man responds. /	200	800

© 2026 The Author(s). Published by WhiteField Academic Press

ect_verb_agreement_2		varies, irregular nouns	The men responds.		
distractor_agreement_relational_noun	Dist-RelN	attractor inside a PP	The essays about art have astounded Theresa. / … has astounded Theresa.	200	800
distractor_agreement_relative_clause	Dist-RC	attractor inside a relative clause	These forks that hadn't alarmed Brett exist. / … exists.	200	800
Total				1,200	4,800

Split. For each paradigm, the 1,000 pair indices were shuffled with Python's random.Random(13) and the first 200 assigned to development, the remaining 800 to test. All rule design used development pairs only. The test pairs were scored once, after the checker's source file had been frozen and its SHA-256 hash recorded; the evaluation script refuses to run if the file changes.

4.3 Baseline models

Kneser–Ney n-gram LMs (trained here). Interpolated Kneser–Ney models [22], [23] of order 3 (KN-3) and 5 (KN-5) with absolute discount D = 0.75 at every order. Training data are four corpora distributed with NLTK [29]: Brown, Project Gutenberg selections, Reuters-21578 and Webtext, together 236,341 sentences and 6,009,745 tokens. Text is lower-cased and tokenised with a regular expression that splits clitics such as n't; words seen fewer than twice map to <unk>, giving 53,081 types. A sentence is scored by its total log-probability including the end-of-sentence symbol.

Released BLiMP model scores. The BLiMP repository provides per-item log-probabilities for the n-gram model, LSTM, Transformer-XL [20] and GPT-2 [19] used by Warstadt et al. [1]. We re-used these scores unchanged. As a check, aggregating them over all 1,000 pairs reproduces each of the 24 published per-paradigm SVA accuracies exactly, and all 4,800 test pairs match our copies of the sentences letter for letter (after undoing tokenisation and HTML escaping). These are the only neural LM results in the paper; we did not re-run any neural LM.

4.4 Proposed model: PACC with agreement-neutralised re-parsing

PACC operates on a spaCy dependency parse [21] (named-entity recognition disabled) and the LemmInflect morphological lexicon [30]. For each sentence it performs the steps below.

(1) Find agreement-bearing verbs. A token qualifies if it is tagged VBZ or VBP; if it is a finite form of be, have or do (including contracted negatives); or, as a fallback for mis-tagged verbs, if it is tagged VB, VBD, NN, NNS, NNP or IN, governs an nsubj/nsubjpass dependent, and its form is a present-tense verb form in the lexicon. A lexical verb with an auxiliary dependent is skipped, because the auxiliary carries agreement.

(2) Read verb number from the surface form. Closed-class auxiliaries use a fixed table (is, was, has, does → singular; are, were, have, do → plural; am → first person). For lexical verbs, a third-person -s form in the lexicon is singular and a bare present form is non-third-singular. Forms that are also past-tense forms of some verb (bore, upset, put) are treated as number-ambiguous and not checked. The tagger's VBZ/VBP decision is used only for forms missing from the lexicon.

(3) Find the subject. The verb's own nsubj, nsubjpass or expl dependent; for an auxiliary, that of its head verb.

(4) Resolve subject number. Personal and demonstrative pronouns use a table. An expletive there takes the number of the post-verbal attribute. A relative pronoun (who, that, which) takes the number of the noun its clause modifies. A subject coordinated with and is plural. A quantity head (lot, lots, number, majority, bunch, couple, plenty) with an of-complement takes the complement's number. Common nouns are looked up in the lexicon, which recognises irregular plurals such as cacti and alumni and the invariant plurals people, police, cattle and clergy; otherwise the parser's morphological Number feature, then the tag, decides.

(5) Compare. am requires I; I and you take non-third-singular forms (and I was); every other subject must match the verb's number. Each mismatch is one violation.

Agreement-neutralised re-parsing (ANR). ANR runs only when the original parse fails to analyse the main clause, that is, when neither the root nor an auxiliary of the root is a checkable finite verb with a subject, and the root is not

© 2026 The Author(s). Published by WhiteField Academic Press

a verb with a subject. Tokens are then visited left to right. Each token that could be a present-tense finite verb is replaced by a number-neutral form (is, are, am → was; has, have → had; does, do → did; a lexical verb → its past tense from the lexicon, skipped if identical to the original). The modified sentence is re-parsed. The first site that the re-parse analyses as the root verb (or the root's auxiliary) with a subject is checked, using the original surface form's number against the subject found in the re-parse. The intuition is that Craig worked with Kristin parses reliably while Craig work with Kristin often does not, and the past tense removes exactly the feature under test.

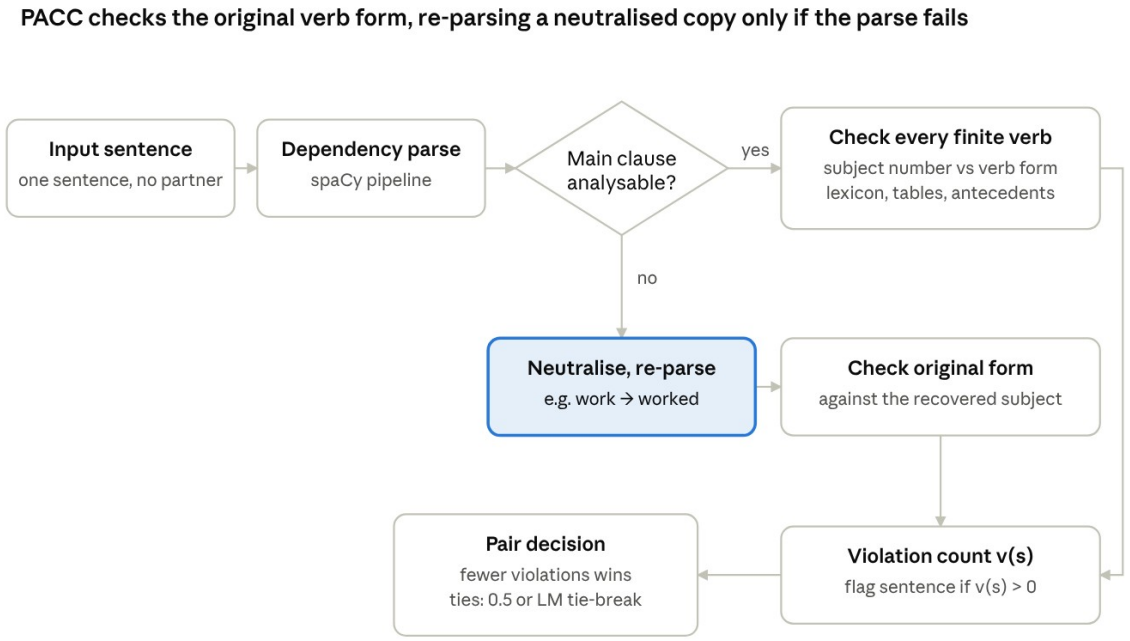


Figure 1 · PACC with agreement-neutralised re-parsing (ANR)

Figure 1. PACC processing of one sentence. ANR (highlighted) runs only when the original parse yields no analysable main clause; the verb form checked is always the original one.

Pair decision and hybrids. For accuracy, a pair scores 1 if $v(s^+) < v(s^-)$, 0 if $v(s^+) > v(s^-)$, and 0.5 on a tie, which is the expected score of a random guess:

$$\text{Acc} = (1/N)\ \Sigma_i\ [\ \mathbb{1}(v(s_i^+) < v(s_i^-)) + \tfrac{1}{2} \cdot \mathbb{1}(v(s_i^+) = v(s_i^-))\ ] \qquad (1)$$

In the hybrid variants "PACC + LM tie-break", ties are instead resolved by the LM's preference, so every pair receives 0 or 1.

Development history. The rules were written in four iterations on the development pairs with the small parser. Strict development accuracy (ties counted as errors) rose from 81.3% for the first version to 92.5% after adding the lexicon and ANR, 93.0% after reading lexical-verb number from the surface form, and 93.2% after excluding lexical verbs with auxiliaries. On the same development pairs the medium and large parsers gave 95.0% and 94.5%, so the medium parser was designated the primary configuration before the test run.

5.0 Experimental Setup

5.1 Systems compared

Table 2. Systems compared.

System	Type	Trained or run here	Notes
KN-3, KN-5	n-gram LM	yes	Section 4.3; 6.0 M training tokens
5-gram, LSTM, Transformer-XL, GPT-2	LMs from [1]	no (released scores)	per-item log-probabilities from the BLiMP repository
PACC-sm / -md / -lg	proposed	yes	spaCy en_core_web_sm, _md, _lg 3.8.0 (CNN-based pipelines)
PACC-trf	proposed	yes, post hoc	spaCy en_core_web_trf 3.8.0

© 2026 The Author(s). Published by WhiteField Academic Press

			(transformer-based pipeline)
PACC-x + KN-5 / + GPT-2 tie-break	hybrid	yes	PACC decides; the LM resolves ties

PACC-md is the primary configuration, fixed on development data. PACC-trf was run after the main test evaluation, with the checker unchanged, to test whether a stronger parser reduces attachment errors; it played no role in any design decision.

5.2 Hyper-parameters and randomness

PACC has no trained parameters. Its only fixed choices are the closed word lists and tables in Section 4.4. The n-gram models use D = 0.75 and a minimum count of 2 for the vocabulary; no tuning was performed. The single random process in the pipeline is the development/test split (seed 13). Bootstrap resampling uses NumPy's default_rng(13).

5.3 Metrics

(a) Pairwise accuracy (Section 4.4), overall and per paradigm; for PACC also the proportions of wins, ties and losses.

(b) Single-sentence detection on the 9,600 test sentences (positive class = unacceptable): precision, recall, F1 and the false-positive rate on acceptable sentences. For LMs, whose raw scores are not detectors, we report ROC-AUC of the per-word log-probability and the F1 obtained with the threshold that maximises F1 on the 2,400 development sentences.

(c) Cost: wall-clock throughput including all re-parses, and peak memory.

5.4 Statistical analysis

Confidence intervals are 95% percentile bootstrap intervals [28] over test pairs with 10,000 resamples. Paired comparisons use the exact McNemar test [25] on per-pair correctness, with PACC ties counted as incorrect so that each system has a binary outcome per pair. The 14 comparisons reported in Tables 4 and 6 are corrected jointly with the Holm procedure [26]; the post-hoc PACC-trf comparison is reported separately and uncorrected.

5.5 Environment

All experiments ran on a Linux cloud workspace with 2 vCPUs (Intel Xeon, 2.80 GHz), 7 GB RAM and no GPU. Software: Python 3.11.15, spaCy 3.8.16, spacy-curated-transformers 0.3.1, PyTorch 2.14.0 (CPU, 2 threads), LemmInflect 0.2.3, NLTK 3.10.3, NumPy 2.4.4, SciPy 1.17.1, scikit-learn 1.8.0. Training KN-5 took 50.9 s with a peak resident memory of 4.3 GB; KN-3 took 19.6 s and 1.1 GB.

6.0 Results

6.1 Pairwise accuracy on the held-out test set

PACC-md, the configuration fixed on development data, prefers the acceptable sentence in 96.5% of the 4,800 test pairs, 9.9 points above the released GPT-2 scores and 44.5 points above the KN-5 model trained here (Table 3). All three CNN-based parsers perform within 0.7 points of each other overall. The gap to the LMs is widest in the two attractor paradigms: on Dist-RelN, PACC-md scores 99.4% against 79.9% for GPT-2, and on Dist-RC 89.4% against 65.9%. Without attractors the gap narrows, and on Reg-1 GPT-2 and PACC-md are close (97.1% vs 97.5%).

Table 3. Pairwise accuracy (%) on the 4,800 test pairs (800 per paradigm). Overall column: mean with 95% bootstrap CI. † Released per-item scores of Warstadt et al. [1], evaluated here on the same test pairs. ‡ Post-hoc run (Section 5.1). PACC rows count ties as 0.5; tie-break rows have no ties.

System	Reg-1	Reg-2	Irr-1	Irr-2	Dist-RelN	Dist-RC	Overall [95% CI]
KN-3	63.4	57.4	54.6	61.9	37.9	35.2	51.7 [50.3, 53.1]
KN-5	63.6	56.9	54.0	65.8	37.2	34.5	52.0 [50.6, 53.4]
5-gram †	75.0	81.4	73.2	87.8	23.0	20.6	60.2 [58.8,

© 2026 The Author(s). Published by WhiteField Academic Press

							61.5]
LSTM †	89.0	82.9	81.2	89.1	74.8	62.1	79.9 [78.7, 81.0]
Transformer-XL †	74.0	84.4	77.5	82.9	77.9	59.2	76.0 [74.8, 77.2]
GPT-2 †	97.1	91.4	93.2	92.2	79.9	65.9	86.6 [85.7, 87.6]
PACC-sm	96.9	98.6	93.6	98.4	98.9	88.2	95.8 [95.2, 96.3]
PACC-md (primary)	97.5	98.9	95.3	98.4	99.4	89.4	96.5 [96.0, 96.9]
PACC-lg	97.6	98.9	96.5	98.9	99.5	86.6	96.3 [95.8, 96.8]
PACC-trf ‡	97.8	99.3	94.9	98.8	99.7	98.0	98.1 [97.7, 98.4]
PACC-md + KN-5 tie-break	97.5	98.6	94.2	98.5	99.1	88.4	96.1 [95.5, 96.6]
PACC-md + GPT-2 tie-break	98.0	99.1	96.6	99.1	99.6	91.4	97.3 [96.9, 97.8]
PACC-trf + GPT-2 tie-break ‡	98.5	99.4	96.9	99.1	99.8	98.6	98.7 [98.4, 99.0]

The KN models trained here are weaker than the released 5-gram (52.0% vs 60.2%), most likely because they saw 6.0 M tokens; the corpus behind the released model is described in [1]. Both n-gram families fall well below chance on the attractor paradigms (20.6–37.9%): a local model sees the attractor next to the verb and prefers the form that agrees with it, which is the wrong one by construction.

PACC's decisions break down into wins, ties and losses as follows: PACC-sm 93.8 / 3.9 / 2.3%, PACC-md 95.1 / 2.9 / 2.1%, PACC-lg 95.1 / 2.4 / 2.5% and PACC-trf 96.9 / 2.3 / 0.8%. A tie means the checker found the same number of violations in both sentences, most often none (Section 6.6).

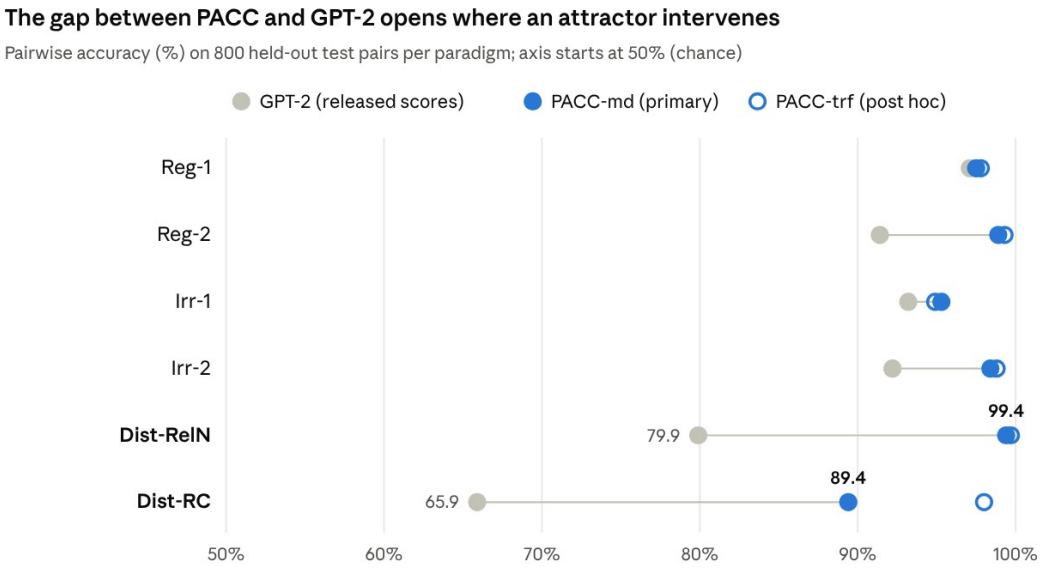


Figure 2 · values from Table 3 (test split, 800 pairs per paradigm); GPT-2 from the released BLiMP scores

Figure 2. Accuracy by paradigm for GPT-2, PACC-md and PACC-trf. Without attractors the three are within 8 points of each other; with an attractor, GPT-2 falls to 79.9% (Dist-RelN) and 65.9% (Dist-RC) while PACC stays above 89%.

6.2 Comparative analysis and significance

© 2026 The Author(s). Published by WhiteField Academic Press

Table 4 gives exact McNemar tests on per-pair correctness, with PACC ties counted as errors. Every PACC hybrid is significantly more accurate than the LM it uses to break ties, and the differences are large relative to their confidence intervals. The medium and large parsers do not differ (85 vs 87 discordant pairs, Holm-adjusted p = 1.0), and both beat the small parser. Using GPT-2 rather than KN-5 to break ties adds 1.25 points (97.3% vs 96.1%). On the 137 pairs that PACC-md leaves tied, GPT-2 picks the acceptable sentence in 78.8% of cases and KN-5 in only 35.0%, so resolving ties with KN-5 lowers accuracy below the chance-level treatment of ties (96.1% vs 96.5%).

Table 4. Exact McNemar tests on the 4,800 test pairs. "A only" / "B only" = pairs correct under one system only. p values Holm-adjusted over the 14 tests in Tables 4 and 6.

System A	System B	A only	B only	Holm-adjusted p
PACC-md + KN-5 tie-break	KN-5	2,154	39	$< 10^{-300}$
PACC-md + KN-5 tie-break	GPT-2 †	577	124	1.1×10^{-69}
PACC-md + GPT-2 tie-break	GPT-2 †	568	55	2.6×10^{-107}
PACC-md + GPT-2 tie-break	PACC-md + KN-5 tie-break	69	9	8.3×10^{-12}
PACC-md	PACC-sm	131	71	1.2×10^{-4}
PACC-lg	PACC-sm	135	73	1.0×10^{-4}
PACC-md	PACC-lg	85	87	1.0

The post-hoc transformer parser is correct on 148 pairs where PACC-md is not and wrong on 59 where PACC-md is correct (exact McNemar $p = 5.2 \times 10^{-10}$, uncorrected).

6.3 Single-sentence detection

As a stand-alone detector, PACC-md flags 95.6% of the 4,800 unacceptable test sentences while flagging 2.96% of the 4,800 acceptable ones (Table 5). LM scores, by contrast, carry almost no single-sentence signal: per-word log-probability separates acceptable from unacceptable sentences with ROC-AUC 0.578 for GPT-2 and 0.510 for KN-5. The F1-maximising threshold chosen on development data therefore degenerates to flagging nearly every sentence; labelling all sentences unacceptable would give precision 50.0%, recall 100% and F1 66.7%. The minimal-pair setting hides this weakness because the partner sentence cancels the lexical and length factors that dominate raw probability.

Table 5. Single-sentence detection on the 9,600 test sentences (positive = unacceptable). FPR = share of acceptable sentences flagged. LM thresholds were tuned on development sentences only.

System	Precision (%)	Recall (%)	F1 (%)	FPR (%)	ROC-AUC
PACC-sm	96.6	94.5	95.6	3.35	–
PACC-md	97.0	95.6	96.3	2.96	–
PACC-lg	96.7	95.8	96.2	3.27	–
PACC-trf ‡	98.6	97.1	97.8	1.42	–
KN-5, per-word log-prob.	50.1	97.9	66.3	97.44	0.510
GPT-2 †, per-word log-prob.	50.3	98.9	66.6	97.85	0.578

6.4 Ablation study

Each PACC component was switched off in turn on the test set with the medium parser (Table 6). Reading lexical-verb number from the surface form matters most: relying on the tagger's VBZ/VBP decision lowers accuracy from 96.5% to 91.1% and nearly quintuples ties (2.9% to 13.9%), because the tagger can assign an ill-formed verb the tag that matches its subject (in development data, *annoy* in *This ladder annoy Paula* was tagged VBZ). ANR is second: without it, accuracy falls to 93.5% and ties triple to 8.7%. The mis-tag fallback, quantity-head

© 2026 The Author(s). Published by WhiteField Academic Press

rule and lexicon-based noun number each contribute 1.2–2.3 points, the last almost entirely on irregular plurals (Irr-1: 95.3% to 89.4%).

Table 6. Ablation of PACC-md on the 4,800 test pairs. Accuracy counts ties as 0.5. "Full only / Ablated only" = discordant pairs in an exact McNemar test on strict wins; p Holm-adjusted jointly with Table 4.

Configuration	Overall (%)	Wins (%)	Ties (%)	Irr-1 (%)	Dist-RC (%)	Full only / Ablated only	Holm-adjusted p
Full PACC-md	96.5	95.1	2.9	95.3	89.4	–	–
− surface-form verb number	91.1	84.1	13.9	90.8	84.7	534 / 9	9.7×10^{-144}
− ANR	93.5	89.2	8.7	92.9	84.7	287 / 4	1.6×10^{-78}
− mis-tag fallback	94.2	90.5	7.5	93.7	87.6	226 / 6	5.4×10^{-58}
− quantity-head rule	94.2	92.8	2.8	91.0	83.4	109 / 1	1.4×10^{-30}
− lexicon noun number	95.3	93.6	3.4	89.4	89.4	72 / 2	2.1×10^{-18}
− relative-pronoun resolution	96.5	95.1	2.8	95.3	89.4	0 / 2	1.0
− coordination rule	96.5	95.1	2.9	95.3	89.4	0 / 0	1.0

Two components have no measurable effect, for reasons that lie in the benchmark rather than the method. BLiMP's SVA paradigms contain no coordinated subjects, and in Dist-RC the verb inside the relative clause is identical in both members of a pair, so checking it cannot separate them; switching the rule off even removes two false alarms. Both rules matter for single sentences in running text, which this benchmark does not test.

6.5 Attractor number and dependency length

In the two distractor paradigms, the head noun's number can be read from the acceptable verb form. Table 7 splits accuracy by that number; 26 Dist-RelN and 69 Dist-RC pairs whose correct verb form does not mark number unambiguously are left out. GPT-2 is markedly less accurate when a plural head is separated from its verb by a singular attractor (67.8% vs 92.3% on Dist-RelN). PACC-md shows no such asymmetry on Dist-RelN (99.5% vs 99.2%), and a small one in the opposite direction on Dist-RC.

Table 7. Accuracy (%) by head-noun number in the distractor paradigms (test pairs).

Paradigm	Head number	n	KN-5	GPT-2 †	PACC-md	PACC-trf ‡
Dist-RelN	singular	376	45.7	92.3	99.2	99.5
Dist-RelN	plural	398	28.9	67.8	99.5	99.9
Dist-RC	singular	327	34.3	72.8	87.8	98.3
Dist-RC	plural	404	34.9	61.6	90.6	97.8

BLiMP also records, for Dist-RC, the linear distance between subject and verb. Across distances of 3, 4 and 5 tokens (n = 143, 342 and 252), PACC-md scores 85.3%, 90.2% and 88.9%, GPT-2 61.5%, 63.2% and 68.7%, and PACC-trf 97.9%, 98.0% and 97.6%. Neither PACC variant degrades with distance in this range, consistent with its reliance on tree structure rather than proximity; longer distances have too few pairs (41, 20 and 2) for a reliable estimate.

6.6 Error analysis

We inspected all 100 test pairs that PACC-md gets wrong (2.1% of pairs) and assigned each to one cause (Table 8).

Table 8. Causes of PACC-md losses on the test set (manual inspection by the author).

Cause	Pairs	Where	Typical case (sentence labelled acceptable, flagged by PACC)

© 2026 The Author(s). Published by WhiteField Academic Press

Attractor attached as subject	56	Dist-RC (all 56)	Some libraries that hadn't remembered Rachel do answer. — the parser makes Rachel the subject of do
Nominal number misread	27	Irr-1 (21), Reg-1 (6)	Those oxen haven't hurt Judy. — oxen is missing from the lexicon and tagged singular
Benchmark label error	17	Reg-1 (7), Reg-2 (4), Irr-2 (4), Irr-1 (1), Dist-RC (1)	Some drivers annoys Derek. — labelled acceptable in BLiMP
Total	100		

Attractor attachment. In each of the 56 Dist-RC losses, the parser attaches the main verb to a noun inside the relative clause as its subject (in one case after tagging the relative-clause verb references as a noun), so the attractor becomes the agreement controller and the acceptable sentence is flagged. The transformer-based parser almost eliminates this pattern: PACC-trf loses only 3 Dist-RC pairs.

Nominal number. Fifteen of the 21 Irr-1 cases involve oxen, which LemmInflect does not list and the tagger labels as a singular noun. The other six are irregular plurals the tagger labels as singular proper nouns, which bypass the lexicon: sentence-initial Octopi (twice), Fungi and Nuclei, and cacti (twice). In Reg-1, the name Carmen is tagged as a plural common noun (three pairs), Politics is tagged plural although used as singular, the invariant plural fish is tagged singular, and in one ANR re-parse Borgias becomes a singular proper noun. These are lexicon and tagging gaps, not failures of the checking logic.

Label errors. In 17 pairs the sentence BLiMP labels acceptable contains an agreement error (The men impresses Steven., A lot of women shocks Anna.), and its partner is the grammatical one. For all 17 pairs the released GPT-2 scores also prefer the sentence labelled unacceptable, which independently supports the reading that the labels, not the systems, are wrong. These pairs set a ceiling below 100% for every system and count against the more accurate ones.

Ties. PACC-md ties on 137 pairs (2.9%). In 114 of them it finds no violation in either sentence, typically because neither the original parse nor ANR yields an analysable finite verb; in 23 it finds violations in both. Ties concentrate in Dist-RC (56) and the irregular-plural paradigms (48).

PACC-trf's 38 losses shift the balance: 15 are among the 17 label errors above, 20 involve nominal number (oxen, octopi, and the ambiguous plurals axes and ellipses), and only 3 fall in Dist-RC. With the stronger parser, lexical coverage and label noise account for most of the remaining error.

6.7 Computational cost

Including all ANR re-parses, PACC processed the 9,600 test sentences at 608, 670 and 684 sentences per second with the small, medium and large CNN pipelines, and at 51 sentences per second with the transformer pipeline, all on 2 CPU cores. The full evaluation, with three parsers loaded in turn, peaked at 1.46 GB of memory. Scoring the 12,000 development and test sentences with KN-5 took 0.6 s once the model was built. PACC needs no GPU and no training; its cost is dominated by parsing.

7.0 Discussion

7.1 What the results show

On controlled English agreement data, an explicit structural check is a stronger detector than LM probability. PACC-md is 9.9 points more accurate than GPT-2 on the same held-out pairs, and the advantage comes almost entirely from the attractor paradigms, the configuration that minimal-pair studies have long identified as the hardest for LMs [2], [3]. Since BLiMP's released scores come from 2019-era models, this comparison says nothing about current large LMs, which may score much higher; what it does show is that a checker with no trained parameters beyond an off-the-shelf parser reaches this level on attractor cases.

The two methods also fail differently. LM errors track surface proximity: n-gram models fall below chance on attractors, and GPT-2 is least accurate when a singular noun separates a plural head from its verb. PACC's errors track parser decisions and lexicon coverage, and are therefore inspectable: every flag names a verb, a subject and the conflicting numbers. The hybrid results show that the two are complementary. When PACC cannot decide,

© 2026 The Author(s). Published by WhiteField Academic Press

GPT-2 is right on 78.8% of the tied pairs, and the combination reaches 97.3% with the medium parser and 98.7% with the transformer parser.

The single-sentence results matter for GEC. Minimal-pair accuracy flatters LM scoring because the partner sentence controls for length and lexical frequency. Without a partner, per-word log-probability barely separates acceptable from unacceptable sentences (ROC-AUC 0.578 for GPT-2), whereas PACC works on one sentence by construction and flags 2.96% of acceptable sentences with the medium parser and 1.42% with the transformer parser.

7.2 Why re-parsing helps

ANR targets a specific mechanism: taggers and parsers learn from well-formed text, so the error token is often the one they misread. Replacing it with a past-tense form removes the agreement feature under test while keeping the sentence's structure, and the parser then recovers the intended analysis. The ablation shows that ANR and surface-form number reading together account for most of PACC's accuracy. The transformer parser's gains on Dist-RC suggest that attachment on the original sentence, rather than the checking logic, is the main remaining bottleneck for the CNN parsers.

7.3 Limitations

(1) Synthetic, English-only data. BLiMP sentences are short and templated. Learner essays and machine-generated text contain other errors near the agreement site, longer sentences and domain vocabulary, all of which stress the parser more. We did not evaluate on learner corpora.

(2) Hand-written rules. The quantity-head list, invariant plurals and pronoun tables were written for English and cover only phenomena seen in development data. Collective nouns, partitives such as half of or none of, either/or coordination and notional agreement are not modelled.

(3) Older LM baselines. The neural LM results are the 2019 scores released with BLiMP; no current large LM was run, because the evaluation environment had no GPU and no access to model repositories.

(4) Small n-gram training corpus. KN-3 and KN-5 were trained on 6.0 M tokens and are weaker than the released 5-gram; they serve as a fully reproducible local baseline, not as the strongest n-gram model possible.

(5) Lexicon dependence. Irregular plurals missing from LemmInflect (oxen) or tagged as proper nouns cause systematic errors.

(6) One annotator. The error categories and the 17 label errors in Table 8 come from inspection by a single annotator, the author.

7.4 Threats to validity

Internal validity. Rule design could overfit the benchmark. We limited this risk by designing on 1,200 development pairs only, freezing the checker with a recorded hash before a single test run, and reporting every configuration tested. The development and test pairs are drawn from the same generator and templates, however, so development-set patterns transfer more easily than they would to new data. PACC-trf and the hybrid variants were added after the design was fixed; no rule was changed for them.

Construct validity. Pairwise preference measures the ability to rank two sentences, not to detect errors; we therefore also report single-sentence metrics. Counting PACC ties as 0.5 is one convention; counting them as errors lowers PACC-md from 96.5% to 95.1%, which is still 8.4 points above GPT-2.

External validity. Results hold for English SVA in the six BLiMP paradigms. Agreement in other languages, other agreement types (determiner–noun, anaphor) and naturally occurring errors remain untested.

Conclusion validity. With 4,800 paired items, the observed differences between PACC and every LM are far larger than their confidence intervals, and Holm correction does not change any conclusion. The 17 benchmark label errors penalise the more accurate systems and bias differences towards zero, so they do not inflate PACC's advantage.

8.0 Conclusion and Future Work

8.1 Conclusion

© 2026 The Author(s). Published by WhiteField Academic Press

We asked whether an explicit, parse-grounded checker detects English subject–verb agreement violations better than language-model scoring. On 4,800 held-out BLiMP minimal pairs, PACC with a medium CNN parser reaches 96.5% pairwise accuracy against 86.6% for the released GPT-2 scores, with the largest gains on attractor constructions, and 98.1% with a transformer parser. As a single-sentence detector it reaches F1 = 96.3% at a 2.96% false-positive rate, a setting in which per-word LM scores are close to uninformative. Two design choices carry most of this performance: reading agreement from the verb's surface form rather than from the tagger, and agreement-neutralised re-parsing, which lets a parser trained on well-formed text analyse the very sentences it would otherwise misread. The remaining errors are mostly parser attachment inside relative clauses, lexicon gaps for irregular plurals, and at least 17 mislabelled pairs in the benchmark itself.

8.2 Future work

(1) Evaluate PACC and ANR on learner corpora with agreement annotations, where errors co-occur and sentences are longer.

(2) Extend ANR to other agreement types in BLiMP (determiner–noun and anaphor agreement) and to languages with richer agreement, using MultiBLiMP [13] and Universal Dependencies parsers.

(3) Compare against current large LMs scored by log-probability, and study hybrids in which the LM is consulted only when the parse is uncertain.

(4) Replace hand-written lists with morphology learned from treebanks, to reduce lexicon gaps such as oxen.

(5) Report the BLiMP label errors identified here to the benchmark maintainers after independent re-annotation.

References

[1] A. Warstadt, A. Parrish, H. Liu, A. Mohananey, W. Peng, S.-F. Wang, and S. R. Bowman, "BLiMP: The Benchmark of Linguistic Minimal Pairs for English," Transactions of the Association for Computational Linguistics, vol. 8, pp. 377–392, 2020, doi: 10.1162/tacl_a_00321.

[2] T. Linzen, E. Dupoux, and Y. Goldberg, "Assessing the ability of LSTMs to learn syntax-sensitive dependencies," Transactions of the Association for Computational Linguistics, vol. 4, pp. 521–535, 2016, doi: 10.1162/tacl_a_00115.

[3] R. Marvin and T. Linzen, "Targeted syntactic evaluation of language models," in Proc. 2018 Conf. Empirical Methods in Natural Language Processing (EMNLP), Brussels, Belgium, 2018, pp. 1192–1202, doi: 10.18653/v1/D18-1151.

[4] K. Gulordava, P. Bojanowski, E. Grave, T. Linzen, and M. Baroni, "Colorless green recurrent networks dream hierarchically," in Proc. 2018 Conf. North American Chapter of the ACL: Human Language Technologies (NAACL-HLT), Vol. 1, New Orleans, LA, USA, 2018, pp. 1195–1205, doi: 10.18653/v1/N18-1108.

[5] J. Hu, J. Gauthier, P. Qian, E. Wilcox, and R. P. Levy, "A systematic assessment of syntactic generalization in neural language models," in Proc. 58th Annu. Meeting of the Association for Computational Linguistics (ACL), 2020, pp. 1725–1744, doi: 10.18653/v1/2020.acl-main.158.

[6] A. Mueller, G. Nicolai, P. Petrou-Zeniou, N. Talmina, and T. Linzen, "Cross-linguistic syntactic evaluation of word prediction models," in Proc. 58th Annu. Meeting of the Association for Computational Linguistics (ACL), 2020, pp. 5523–5539, doi: 10.18653/v1/2020.acl-main.490.

[7] B. Newman, K.-S. Ang, J. Gong, and J. Hewitt, "Refining targeted syntactic evaluation of language models," in Proc. 2021 Conf. North American Chapter of the ACL: Human Language Technologies (NAACL-HLT), 2021, pp. 3710–3723, doi: 10.18653/v1/2021.naacl-main.290.

[8] M. Finlayson, A. Mueller, S. Gehrmann, S. Shieber, T. Linzen, and Y. Belinkov, "Causal analysis of syntactic agreement mechanisms in neural language models," in Proc. 59th Annu. Meeting of the ACL and 11th Int. Joint Conf. Natural Language Processing (ACL-IJCNLP), Vol. 1, 2021, pp. 1828–1843, doi: 10.18653/v1/2021.acl-long.144.

[9] J. Wei, D. Garrette, T. Linzen, and E. Pavlick, "Frequency effects on syntactic rule learning in transformers," in Proc. 2021 Conf. Empirical Methods in Natural Language Processing (EMNLP), 2021, pp. 932–948, doi: 10.18653/v1/2021.emnlp-main.72.

[10] J. Vamvas and R. Sennrich, "On the limits of minimal pairs in contrastive evaluation," in Proc. Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, Punta Cana, Dominican Republic, 2021, pp. 58–68, doi: 10.18653/v1/2021.blackboxnlp-1.5.

[11] J. Hu and R. Levy, "Prompting is not a substitute for probability measurements in large language models," in Proc. 2023 Conf. Empirical Methods in Natural Language Processing (EMNLP), Singapore, 2023, pp. 5040–5060, doi: 10.18653/v1/2023.emnlp-main.306.

[12] J. Hu, K. Mahowald, G. Lupyan, A. Ivanova, and R. Levy, "Language models align with human judgments on key grammatical constructions," Proceedings of the National Academy of Sciences, vol. 121, no. 36, Art. no. e2400917121, 2024, doi: 10.1073/pnas.2400917121.

[13] J. Jumelet, L. Weissweiler, J. Nivre, and A. Bisazza, "MultiBLiMP 1.0: A massively multilingual benchmark of linguistic minimal pairs," Transactions of the Association for Computational Linguistics, vol. 14, pp. 193–216, 2026, doi: 10.1162/tacl.a.600.

[14] A. Warstadt, A. Singh, and S. R. Bowman, "Neural network acceptability judgments," Transactions of the Association for Computational Linguistics, vol. 7, pp. 625–641, 2019, doi: 10.1162/tacl_a_00290.

© 2026 The Author(s). Published by WhiteField Academic Press

[15] C. Bryant, Z. Yuan, M. R. Qorib, H. Cao, H. T. Ng, and T. Briscoe, “Grammatical error correction: A survey of the state of the art,” Computational Linguistics, vol. 49, no. 3, pp. 643–701, 2023, doi: 10.1162/coli_a_00478.

[16] K. Omelianchuk, V. Atrasevych, A. Chernodub, and O. Skurzhanskyi, “GECToR – Grammatical error correction: Tag, not rewrite,” in Proc. 15th Workshop on Innovative Use of NLP for Building Educational Applications (BEA), 2020, pp. 163–170, doi: 10.18653/v1/2020.bea-1.16.

[17] S. Rothe, J. Mallinson, E. Malmi, S. Krause, and A. Severyn, “A simple recipe for multilingual grammatical error correction,” in Proc. 59th Annu. Meeting of the ACL and 11th Int. Joint Conf. Natural Language Processing (ACL-IJCNLP), Vol. 2, 2021, pp. 702–707, doi: 10.18653/v1/2021.acl-short.89.

[18] H. B. Hashemi and R. Hwa, “An evaluation of parser robustness for ungrammatical sentences,” in Proc. 2016 Conf. Empirical Methods in Natural Language Processing (EMNLP), Austin, TX, USA, 2016, pp. 1765–1774, doi: 10.18653/v1/D16-1182.

[19] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” OpenAI, Technical report, 2019. [Online]. Available: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[20] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, “Transformer-XL: Attentive language models beyond a fixed-length context,” in Proc. 57th Annu. Meeting of the Association for Computational Linguistics (ACL), Florence, Italy, 2019, pp. 2978–2988, doi: 10.18653/v1/P19-1285.

[21] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, “spaCy: Industrial-strength natural language processing in Python,” 2020, Zenodo, doi: 10.5281/zenodo.1212303.

[22] S. F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” Computer Speech & Language, vol. 13, no. 4, pp. 359–393, 1999, doi: 10.1006/csla.1999.0128.

[23] R. Kneser and H. Ney, “Improved backing-off for M-gram language modeling,” in Proc. 1995 Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP), vol. 1, 1995, pp. 181–184, doi: 10.1109/ICASSP.1995.479394.

[24] M. Loem, M. Kaneko, S. Takase, and N. Okazaki, “Exploring effectiveness of GPT-3 in grammatical error correction: A study on performance and controllability in prompt-based methods,” in Proc. 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA), Toronto, Canada, 2023, pp. 205–219, doi: 10.18653/v1/2023.bea-1.18.

[25] Q. McNemar, “Note on the sampling error of the difference between correlated proportions or percentages,” Psychometrika, vol. 12, no. 2, pp. 153–157, 1947, doi: 10.1007/BF02295996.

[26] S. Holm, “A simple sequentially rejective multiple test procedure,” Scandinavian Journal of Statistics, vol. 6, no. 2, pp. 65–70, 1979.

[27] K. Bock and C. A. Miller, “Broken agreement,” Cognitive Psychology, vol. 23, no. 1, pp. 45–93, 1991, doi: 10.1016/0010-0285(91)90003-7.

[28] B. Efron, “Bootstrap methods: Another look at the jackknife,” The Annals of Statistics, vol. 7, no. 1, 1979, doi: 10.1214/aos/1176344552.

[29] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. Sebastopol, CA, USA: O’Reilly Media, 2009, ISBN 978-0-596-51649-9.

[30] B. Jascob, “LemmInflect: A Python module for English lemmatization and inflection,” version 0.2.3, software. [Online]. Available: https://github.com/bjascob/LemmInflect

© 2026 The Author(s). Published by WhiteField Academic Press. This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited. *Authors retain copyright.*

INTERNATIONAL JOURNAL OF ADVANCED COMPUTING AND MACHINE INTELLIGENCE

Peer-Reviewed · Open Access · Published by WhiteField Academic Press
Noida, India

Scan to visit

www.ijacmi.org
editor@ijacmi.org
submissions@ijacmi.org

Visit our website — www.ijacmi.org

© 2026 The Author(s). Published by WhiteField Academic Press