OPEN ACCESS

GLODAL Emerging Educational Technologies

Emerging Educational Technologies, 2026, 1,
260105
https://doi.org/10.65553/EET.260105

Vibe Coding for Corpus-Based Language Research and Learning: From Working Tools to Warranted Claims

Wei Lun Wong[a]

[a]Faculty of Education, Universiti Kebangsaan Malaysia

Submission date: May 21, 2026; Revision date: October 25, 2026; Online: November 20, 2026

Abstract

Vibe coding allows researchers and teachers to build software through natural language interaction with generative artificial intelligence. It could widen access to corpus analysis and support tools designed for specific learning contexts. However, easier software creation also transfers methodological choices into prompts, generates code and interface defaults. Crosthwaite and Özer (2026) have begun to specify how such development should be documented, most fully in vibecoding-as-method workflow. However, a documented process does not establish what a finished tool entitles its users to claim. This conceptual article examines that gap through a purposive interpretative synthesis of corpus linguistics, data-driven learning and current corpus-AI research. It argues that a functional tool provides insufficient evidence of a valid linguistic analysis or an effective learning activity and proposes three linked forms of validation: computational correctness, linguistic validity, and pedagogical validity. Each requires evidence appropriate to the claims being made. Each leaves a specific claim unsupported. The framework is illustrated through a published vibe-coded instrument, VetLingua, which shows where a carefully documented development process still leaves validation evidence to be gathered. The article also outlines a research agenda that compares complete workflows and tests independent learning beyond immediate task performance. Vibe coding could become a useful method in corpus research and education when institutions support transparent development, independent evaluation and continued maintenance.

Keywords: vibe coding; generative artificial intelligence; corpus linguistics; data-driven learning; methodological validity; validation; research instruments

Copyright © 2026 The Authors
Wei Lun Wong, corresponding author: colinw@ukm.edu.my

Introduction

Vibe coding changes who can build a corpus tool. The term was coined for a style of programming in which a developer describes what software should do and accepts the generated code with little or no inspection (Karpathy, 2025). In research settings, a researcher can describe a concordancer, a frequency analysis or a learning interface in ordinary language and ask an AI system to generate the software. This possibility matters in applied linguistics, where programming knowledge can restrict instrument development. Crosthwaite and Özer (2026) demonstrate a structured approach through VetLingua, a tool for veterinary language learning. The central issue is how researchers establish control over the instruments that AI helps them create.

That issue extends an existing debate about corpora and generative AI. Corpus analysis can identify patterns in a defined collection of texts. AI can explain those patterns, generate practice materials and simplify interaction with software. These functions may complement each other, but they produce different kinds of evidence (Crosthwaite & Baisa, 2023; Mizumoto, 2023). Vibe coding adds another function. AI becomes involved in building the instrument through which language evidence is selected and analysed. Errors at this stage can affect every later interpretation, even when the interface appears clear and the output looks plausible.

Four strands of recent work bear on this issue. Ma et al. (2024; 2026) connect corpus use to teacher knowledge and pedagogical practice. Cheung and Crosthwaite (2025) and Crosthwaite and Baisa (2024) address corpus tools, AI integration and instrument development. Pérez-Paredes (2026) and Pérez-Paredes et al. (2025) develop a critical account of corpus and AI literacies. Wong et al. (2026) map the wider intersection of corpus linguistics and AI. These contributions provide different foundations for the present argument. Their relevance should be assessed through what each study establishes. Corpus pedagogy, AI literacy and software development should not be treated as equivalent research areas.

The closest precedent is Crosthwaite and Özer (2026) methods tutorial, which reframes vibe coding as vibecoding-as-method. It is an eight-stage workflow, situated within design-based research, in which prompts are treated as methodological decisions that must be documented, justified and audited. That tutorial describes itself as a first articulation and it limits its knowledge claims to the development cycle. Its deployment, logging and evaluation stages are presented as design intentions. The present article begins where that account stops. A documented process shows how an instrument was built. It does not show that the instrument computes what it is specified to compute, that its outputs support the linguistic claims made from them, or that learners who use it develop abilities they retain. These are questions about evidence, and they arise after the development record is complete.

The article makes three contributions. First, it separates three claims that discussions of AI-built tools tend to merge and it identifies the evidence each requires and the inference each cannot support. Second, it distinguishes how a tool is produced from whether AI continues to act when the tool is used because the two carry different threats to reproducibility. Third, it applies the resulting framework to a published instrument and derives a research agenda that can be tested, including comparisons of complete development workflows and measures of learning with assistance withdrawn.

This conceptual article offers an interpretative narrative synthesis of current research and selected foundational studies. It argues that vibe coding can reduce the effort required to create a tool while increasing the need to justify its analytical decisions. Following this argument, the review first clarifies the relation between vibe coding and corpus inquiry. It then examines the main methodological and pedagogical problems. The later sections propose a framework for validation, apply it to a published case and set out a research agenda. The future trend considered here is conditional. Its value will depend on how tool creation is connected to

credible evidence and independent learning.

Approach and scope

This article is conceptual. Its synthesis is purposive and interpretative and it makes no claim to exhaustive coverage. It draws on three bodies of work, each of which supplies a standard against which a vibe-coded tool can be judged. Foundational corpus linguistics supplies standards for sampling, counting and interpretation (Biber, 1993; Gries, 2008; Kilgarriff, 2005). Research on data-driven learning supplies evidence about what corpus use contributes to learning, and under which conditions (Boulton & Cobb, 2017; Boulton & Vyatkina, 2021). Studies of corpus-AI integration published since 2023 supply the current tools, claims and cautions (Ren et al., 2026; Zhang et al., 2025). Work on vibe coding outside applied linguistics is used more selectively to locate the practice and its known reliability problems (Alqheedan & Alzughaibi, 2026; Elgendy et al., 2026; Ouyang et al., 2025).

Sources are cited for what they establish. A study of teacher self-efficacy is not treated as evidence of pupil learning, and a meta-analysis of corpus use is not treated as validation of a new way of building corpus software. The unit of analysis is the claim made from a tool, not the tool itself. Throughout, "vibe-coded tool" refers to software produced mainly through natural language interaction with a generative model, whether or not the finished program calls a model when it runs. The article reports no new empirical data. Its framework is a proposal and its illustrative case rests on a published account, not on independent testing of the instrument.

Background: vibe coding and corpus inquiry

Understanding vibe coding in corpus-based language research and learning

Vibe coding refers here to developing software through repeated natural language interaction with an AI system. The user describes a function, inspects the result and requests changes. This usage is broader than the original one. Karpathy (2025) described a practice in which the code itself is largely ignored. Research use cannot adopt that stance, because the code is the instrument. Work in information systems has accordingly begun to treat reliability and governance as open problems for vibe coding (Elgendy et al., 2026). In health informatics, Alqheedan and Alzughaibi (2026) propose a conceptual framework for judging the appropriateness, quality and safety of vibe-coded clinical tools and illustrate it with published cases. No comparable account exists for language research. Crosthwaite and Özer (2026) adapt this process into a documented workflow for research instruments. Their contribution concerns methodological design and a worked development case. It should be read as a proposal requiring further validation across contexts. A successful prototype establishes that a function can be built. Claims about analytical accuracy, usability or learning require additional evidence.

The phrase "vibe-coded AI" can obscure this distinction. A language researcher usually uses an existing AI model to generate software. The researcher may build a corpus application without developing a new AI model (Wong & Fong, 2025). Moreover, the resulting application does not necessarily use AI during analysis. A generated program can count words through fixed rules. Another program may ask a language model to label texts or explain concordance lines. These designs carry different uncertainties. Researchers must identify where AI acts, which operations remain fixed and which outputs can vary between runs.

Corpus linguistics provides a demanding test of such development. A corpus is a collection assembled for a stated purpose, with identifiable sources and a defined sampling logic (McEnery & Hardie, 2012; Sinclair, 2005). Biber (1993) shows why representativeness depends on the population and the dimensions of variation being studied. A large collection of accessible texts cannot automatically represent a language, a discipline or a learner group. Vibe coding can accelerate collection and processing, but it cannot establish the relation between a sample and a research claim. That relation remains a theoretical and empirical responsibility.

The same principle applies to quantitative analysis. Frequencies depend on what counts as a

word and how the corpus is segmented. Their interpretation also depends on dispersion, or how occurrences are distributed across texts (Gries, 2008). Repeated uses by one writer may create a high frequency without indicating a widespread pattern. Kilgarriff (2005) challenges the assumption that language data behave like simple random observations. Together, these arguments show why convenient counts need careful interpretation. An AI-generated script can reproduce a formula accurately while encouraging an unsuitable comparison.

Automated analysis already has an established history in language research. Lu (2010) system for syntactic complexity evaluates specified measures against learner writing data. Research on lexical richness also links the interpretation of indices to the language performance being investigated (Lu, 2012). These studies offer an important standard for vibe-coded tools. Their value rests on defined measures and evaluation, rather than automation alone. A new application should explain how it defines a construct and where its measurements have been checked. Familiar labels such as "complexity" or "proficiency" cannot substitute for that explanation.

Vibe coding is also relevant to data-driven learning (DDL), where learners investigate patterns in language data. The central learning activity is the interpretation of evidence. Access to a computer is only one possible means of organising that activity. Boulton's work on paper-based DDL demonstrates why corpus learning need not depend on direct software use (Boulton, 2010). Frankenberg-Garcia (2014) further shows that corpus examples for comprehension and production serve different purposes. Therefore, an interface becomes educationally meaningful when its functions support the intended learning process. Greater technical sophistication is not a sufficient educational justification.

The evidence for corpus learning supports cautious development. Boulton and Cobb (2017) identify substantial learning benefits across the studies reviewed. Those findings support corpus use under the conditions represented in that literature. They do not validate every corpus application or a new method of generating software. Boulton and Vyatkina (2021) show how DDL has developed over time, while Pérez-Paredes (2022) examines its spread within CALL research. These reviews make a cumulative research base available. Vibe coding should extend that base through comparable evidence, instead of treating technical novelty as a fresh starting point.

The practical gap remains significant. Chambers (2019) examines the distance between corpus research and ordinary teaching practice. CorpusMate responds to interface difficulties through a tool designed around disciplinary learning needs (Crosthwaite & Baisa, 2024). Charles (2024) discusses the benefits and demands of learners constructing their own corpora for academic writing. These contributions suggest that access involves more than obtaining software. Teachers and learners need relevant texts, manageable tasks and enough support to interpret findings. Vibe coding may help tailor those elements, but it can also transfer design and maintenance work to people who already have limited time.

Ma et al. (2024) connect corpus pedagogy to language knowledge, technology, teaching practice and context. This account explains why the ability to request a feature is only a small part of pedagogical design. A teacher must decide what learners should notice, how much context they need and what they can reasonably infer. Ma et al. (2026) relate corpus training and knowledge to student teachers' self-efficacy. Confidence is relevant to adoption, but it is a different outcome from pupils' language development. Applying this work to vibe coding therefore requires evaluation of both teacher preparation and classroom learning.

Recent tools make the convergence between corpora and AI more concrete. CorpusChat integrates corpus linguistics and generative AI for academic writing, while ChatAI addresses the integration of AI with corpus-based learning (Anthony, 2025; Cheung & Crosthwaite, 2025). Their relevance lies in how they connect language evidence to interaction and support. They should not automatically be classified as vibe-coded tools. Corpus-AI integration concerns what an application does. Vibe coding concerns how an application is developed. Keeping these questions separate allows researchers to evaluate the contribution of AI-

supported development without attributing every benefit of a corpus interface to its production method.

Pérez-Paredes (2026) positions corpus literacy as a basis for transparent AI-mediated learning. Pérez-Paredes et al. (2025) examine this relation through workshops involving corpus consultation and ChatGPT. Their findings support explicit attention to reliability and critical engagement. Vibe coding extends that concern to instrument creation. Users need to examine the choices built into a tool, as well as the answers produced by it. The important development is therefore a possible redistribution of methodological authority. Researchers and teachers gain more influence over software design, but must also account for decisions previously embedded in established tools.

From documented process to warranted claims

Crosthwaite and Özer (2026) provide the most developed account of vibe coding as a research practice in applied linguistics. Their workflow runs from problem specification and corpus preparation through feature operationalisation, prompt design, and interface and backend generation, to deployment, logging and evaluation. Several of its principles are adopted here without modification. Corpus preparation should remain outside the AI interface, so that generated text does not enter the evidence base. Constructs should be grounded in theory and checked by the researcher against the corpus before they are written into prompts. Prompts, model versions and settings should be recorded. Earlier functions should be retested after each change. The tutorial also states four conditions under which a vibe-coded system can count as a research instrument: theoretically precise operationalisation, consistent and documented elicitation conditions, a replicable record of prompt iterations, and openly available code.

These conditions concern how an instrument is produced and recorded. They are necessary. They are not sufficient for the claims researchers and teachers eventually make. A prompt history can be complete while the retrieval function it produced miscounts across document boundaries. A construct can be defined with care while the tool's generated explanations misdescribe the concordance lines they accompany. A tool can meet all four conditions and still leave learners no better able to consult evidence without it. The tutorial does not claim otherwise. It positions vibecoding-as-method as a pre-deployment methodology and calls for validation across further contexts. The question it leaves open is what that validation should consist of.

Two further distinctions shape the answer. The first is between verification and validation, a distinction familiar from software engineering. Verification asks whether a program does what its specification says. Validation asks whether the specification, and the interpretation placed on its output, are appropriate to the purpose. Kane (2013) frames the second question as an argument in which each inference from an observed score to a claim needs its own warrant. The second distinction is between AI in development and AI in operation. Where a model only writes the code, the finished program can be tested once and will then behave consistently. Where a model is called each time the tool is used, as in tools that generate explanations, exercises or feedback, the behaviour under test can change between runs and between model versions. Validation evidence for the second kind of tool has a shorter life and must be renewed.

Critical issues in vibe-coded corpus research and learning

Convenience, provenance and the integrity of evidence

The first problem concerns the relation between convenience and evidence. A clear interface can make outputs appear settled, especially when charts and explanations arrive together. However, users may understand the displayed result better than the process that produced it. Messeri and Crockett (2024) explain how AI can create an illusion of understanding in scientific work. In corpus research, this risk arises when fluency or visual polish conceals sampling choices, preprocessing rules or uncertain labels. The response should be to make these choices inspectable. A general warning to "use AI critically" provides little protection unless users can identify what requires checking.

Corpus provenance is central to that response. Researchers need to know where texts came from, why they were selected and what happened to them during preparation. Crosthwaite and Baisa (2023) emphasise the advantages of identifiable corpus evidence over opaque model output. Vibe coding does not remove this advantage, but poorly specified software can weaken it. A script may merge documents, remove metadata or silently exclude difficult files. The final dataset can then differ from the intended sample. Researchers should compare the prepared corpus with the source collection and account for exclusions before interpreting a frequency list. Crosthwaite and Özer (2026) make the same point procedurally, by requiring that corpus preparation take place outside the AI interface. The requirement protects the source texts. It does not by itself show that the generated ingestion code preserved them, which is why the prepared corpus still needs to be checked against its source.

Generated language creates a related problem. Synthetic examples can support practice or test a program, but they need clear labels and a purpose appropriate to their origin. Shumailov et al. (2024) show how recursive training on generated data can distort model distributions. That finding concerns model training and should not be transferred directly to every classroom use of generated examples. Its relevance here is more specific. Generated material cannot be assumed to preserve the variation found in observed language. Mixing it with authentic data without separate reporting can change the object that a corpus study claims to investigate.

Automated correction can have a similar effect. If a program rewrites learner texts before analysis, some features of learner language may disappear. An apparent improvement in corpus quality can become a loss of evidence about development or variation. Barrot (2023) discusses both the potential and the pitfalls of ChatGPT in second language writing. For corpus research, the critical decision is whether AI output is part of the phenomenon being studied or an intervention in the data. Original texts, edited texts and generated texts should remain distinguishable. Otherwise, the study cannot explain whose language it has measured.

Prompts, specification and reproducibility

A second problem concerns prompts as methodological decisions, the central claim of Crosthwaite and Özer (2026). Evidence for how much a prompt can change an outcome comes from a neighbouring line of work. Crosthwaite et al. (2026) compared 145 student essays from the LOCNESS corpus with two sets of 145 GPT-generated essays on the same topics. Under a general essay prompt, the generated texts contained fewer transitions, hedges, self-mentions, attitude markers and engagement resources than the student writing. Under a prompt that explicitly requested such features, several of them rose above both the baseline and the human frequencies, and they were often realised in more formulaic ways. Their study cautions against interpreting one elicitation condition as a stable property of AI writing. The implication for vibe coding is related, although it requires its own evidence. Instructions such as "remove noise" or "find academic phrases" leave important choices open. The model may decide what noise or a phrase means. Researchers should specify those choices before development and report the decisions that remain uncertain.

Natural language instructions can also conceal technical disagreements. A request to "compare common phrases" might lead to raw frequencies, normalised counts or a statistical comparison. Each answer addresses a different question. Changing the wording until a result

looks convincing may alter the analysis along with the interface. Prompt refinement must therefore be tied to a written specification. That specification should define the unit of analysis, the comparison and the expected output. The record should show which changes corrected an error and which changed the research question. Otherwise, a development history may become an undocumented sequence of analytical revisions.

Reproducibility requires more than saving prompts. Repeating a prompt may generate different code because the model or development environment has changed. A fixed program can produce stable results even when the process that created it was variable. Conversely, a program that calls an AI model during annotation may remain variable after its code is saved. Research teams should preserve the final code, dependencies, settings and permitted data materials. They should also record model details where AI remains part of analysis. Marsden et al. (2018) make replication a substantive concern in language research. Vibe coding needs to meet that concern through reproducible operations and transparent limits. The difficulty is already documented for generated code. In a study of 829 code generation problems, Ouyang et al. (2025) found that identical requests to ChatGPT often returned programs that behaved differently, and that setting the temperature to zero reduced this variation without removing it.

From correct computation to valid interpretation

A third problem is the confusion between correct computation and valid interpretation. A program may count every occurrence accurately while counting the wrong feature for the research question. Kane (2013) locates validity in the justification of interpretations and uses of scores. Although his account concerns assessment, its reasoning can inform corpus instrument evaluation. The researcher must connect an observed output to the claim made from it. For example, an index labelled "academic style" requires an account of which language features it measures and why they support that interpretation. A working calculation cannot supply this argument on its own.

Annotation makes this problem especially visible. Mizumoto and Eguchi (2023) evaluate AI-assisted essay scoring against an established learner corpus. Their study illustrates the need to investigate performance for a specified task and dataset. It cannot validate a later model, prompt or classroom population automatically. Likewise, a vibe-coded tool that labels stance must be evaluated using a clear definition and independently adjudicated examples. Agreement between repeated AI runs establishes a form of consistency. It does not show that the labels capture the intended linguistic function. Consistent misclassification remains a threat to interpretation.

Quantification should also preserve the possibility of close reading. Baker et al. (2008) demonstrate how corpus analysis and critical discourse analysis can inform one another. An AI-generated summary can help locate a pattern, but it may conceal ambiguity or disagreement within the corpus. Researchers should inspect original contexts, including examples that challenge the dominant interpretation. The issue is particularly important when a tool turns a word association into a claim about ideology or social identity. Such claims require contextual evidence and theoretical reasoning. A model's fluent explanation is a proposed interpretation that must remain open to revision.

Learner reasoning and pedagogical design

The pedagogical problem is different but closely connected. Easier interaction can reduce the effort of searching a corpus. It can also remove the reasoning that gives DDL its educational value. O'Sullivan (2007) treats corpus consultation as part of a process of literacy development. Mizumoto (2023) similarly connects resource use to metacognitive regulation. These accounts suggest that a learner should decide what evidence is needed and whether it

supports a claim. An application that supplies an answer before the learner examines examples may improve immediate task completion while limiting opportunities to develop those decisions.

This does not mean that every difficulty should be retained. Learners can spend excessive effort navigating syntax, correcting search errors or managing irrelevant results. Risko and Gilbert (2016) explain how cognitive offloading can reduce task demands, while also depending on judgements about one's abilities. For corpus learning, support should reduce avoidable technical effort while preserving the reasoning targeted by the activity. A tool might simplify a search but still ask learners to compare contexts and explain exceptions. The appropriate amount of assistance depends on the task and the learner. It should be investigated through learning outcomes, rather than inferred from ease of use.

Process evidence can help determine what learners actually do. Pérez-Paredes et al. (2011) use logs to investigate corpus consultation under different guidance conditions. Their later work examines learners' search patterns during corpus-based activities (Pérez-Paredes et al., 2012). These approaches remain relevant to AI-mediated interfaces, where a completed text may conceal very different learning processes. Researchers need to examine whether learners inspect evidence, revise a query or simply accept an explanation. Logs can record these actions, but their meaning still requires interpretation. A large number of clicks cannot establish attention, understanding or transfer without supporting evidence. Such evidence has to be planned. Crosthwaite and Özer (2026) report that VetLingua's analytics currently record access and traffic but not queries or interaction sequences and they conclude that research-grade logging must be designed into an instrument from the outset.

Current corpus-AI writing studies offer useful directions, but their findings need careful attribution. Dong and Wang (2025) investigate a DDL-GenAI teaching mode. Wang et al. (2025) examine corpus-driven and AI-assisted approaches alongside learner variables. These studies address instructional combinations. They do not isolate whether generating the software through vibe coding improves learning. Outcomes could depend on corpus selection, feedback, teacher guidance or time on task. A study of vibe coding should therefore define the contribution being tested. Development efficiency and instructional effectiveness are legitimate outcomes, but they require different comparisons.

Teachers, access and accountability

Teacher knowledge is another condition that cannot be delegated to software. Ma's work on corpus pedagogy shows why technology must connect to content, pedagogy and context (Ma et al., 2024). Celik (2023) extends this concern to teachers' professional knowledge for ethical AI integration. These perspectives challenge the assumption that removing a programming barrier also removes the need for preparation. Teachers still need to recognise misleading examples, unsuitable difficulty and unsupported claims. They also need a feasible route to technical assistance. Requiring every teacher to maintain a custom application could increase workload and deepen differences between well-supported and under-supported institutions.

Access must therefore be assessed beyond the ability to enter a prompt. Wong et al. (2026) map 162 selected open-access records from Scopus between 2020 and mid-2025. Their review identifies thematic and geographical patterns in a bounded publication sample. It cannot establish the prevalence of vibe coding or its classroom effectiveness. Its value for this discussion is to make research visibility and uneven participation objects of inquiry. Language communities with limited digitised data or technical support may gain less from rapid tool generation. A future research agenda should examine who can create, validate and sustain corpus applications.

The professional question concerns who remains accountable for those applications. Kern (2021) connects digital literacy to the social practices through which technologies are used. Ng

et al. (2021) include understanding, evaluation and ethical concerns within AI literacy. Vibe coding brings these responsibilities into software production. Researchers need decisions about data access, consent where required and the handling of identifiable learner texts. They also need a plan for failed analyses and abandoned tools. A system that is easy to create but difficult to inspect or maintain can shift costs to students, colleagues and future researchers.

A framework for validating vibe-coded corpus tools

Three forms of validation

The future value of vibe coding should be judged through the claims a tool supports. This article proposes three linked forms of validation. Computational correctness concerns whether the program carries out its specified operations. Linguistic validity concerns whether the data and measures justify the language claim. Pedagogical validity concerns whether the activity supports the intended learning. These checks adapt established concerns to the development of corpus instruments. They are a conceptual framework for investigation, rather than an empirically validated model. Table 1 identifies the evidence each check would require and the conclusion it cannot establish by itself.

Computational checking should begin with a small corpus whose expected results are known. Researchers can specify how the tool should handle punctuation, empty files and document boundaries. They should then compare its output with those expectations and, where appropriate, an established implementation. Different tools may use different definitions, so disagreement requires investigation. AI can help write tests or suggest difficult cases. However, the same model should not supply the only definition of correctness for code it generated. The reference specification and expected results need an independent basis. Tests should be repeated after changes that affect analysis. In software terms these are known-answer tests, and repeating them after each change is regression testing. Crosthwaite and Özer (2026) recommend retesting earlier functionality after each prompt cycle. The addition proposed here is that the expected results be fixed in advance and derived independently of the model.

Table 1. Evidence required to validate corpus tools

Form of validation	Evidence required	What it does not establish
Computational correctness	Known-answer examples, checks of document boundaries and comparisons using the same analytical definitions.	That the measure captures the intended linguistic construct.
Linguistic validity	A justified corpus sample, defined constructs, inspection of original contexts and analysis of alternative explanations.	That using the tool leads to language learning.
Pedagogical validity	Appropriate comparison tasks, delayed assessment, evidence of learner reasoning and transfer with reduced assistance.	That every corpus, model or learner group will show the same benefit.

Note. This conceptual framework is proposed in the present article. It requires empirical evaluation.

Linguistic validation requires the next step in the argument. Consider a hypothetical application that extracts lexical bundles from student essays. Biber et al. (2004) show that bundles vary across instructional and academic registers. Gries (2008) makes dispersion relevant to frequency interpretation. Following these insights, the application should expose its

tokenisation rules, thresholds and distribution across writers. It should preserve document boundaries and provide the original contexts. A technically correct list may still be unsuitable for comparing proficiency groups if their tasks, disciplines or text lengths differ. Researchers must justify the comparison before assigning developmental meaning to the results.

The same example shows why validation should be iterative. An unexpected bundle may reveal a preprocessing error. It may also identify a real phrase concentrated in one assignment. The first problem requires a software correction. The second requires an interpretation of corpus composition. Treating every unexpected result as a bug would remove potentially important evidence. Treating every output as a discovery would preserve avoidable errors. Researchers should document how they resolved each case and retain the relevant data. Such decisions make the analysis reviewable and help later researchers distinguish a methodological choice from a technical failure.

Pedagogical validation requires evidence of learning beyond the immediate interface. A hypothetical writing tool could retrieve authentic concordance lines and offer an AI explanation in a separate panel. Learners could first propose a pattern, then inspect the explanation and test it against further examples. The design should keep corpus evidence accessible and identify generated content. Pérez-Paredes (2026) provides a foundation for this approach. Assessment should include a later task with reduced assistance or a new topic. That comparison would help establish whether learners acquired a strategy they can use independently, rather than only completing the supported task more successfully.

The three forms are ordered and the order matters for interpretation. Linguistic claims cannot be stronger than the computation beneath them, and a learning gain attributed to corpus evidence presupposes that the evidence shown to learners was sound. The dependence does not run in reverse. A tool can pass every computational test and support no defensible linguistic claim and a linguistically sound tool can still fail to help learners. Reports of vibe-coded tools should state which of the three claims they make and present the corresponding evidence. A development report that makes only the first claim is legitimate, provided it is not read as supporting the other two. Where a tool calls a language model in operation, each form of evidence should also carry the model version and date, since a later model can invalidate earlier checks.

An illustrative application: VetLingua

The framework can be illustrated with the instrument that Crosthwaite and Özer (2026) describe. VetLingua is a vibe-coded tool for veterinary academic English built on the Veterinary Medicine Corpus. It offers a writing assistant, vocabulary practice and writing feedback, and users can query it through preset buttons or free conversation. The reading that follows relies only on the published tutorial. It is not an evaluation of the tool and it is offered to show what the framework asks of a well-documented case. Table 2 summarises the comparison.

On computational correctness, the tutorial reports iterative debugging and one failure in detail: queries in English were answered in Spanish. The fault was reported by users and then corrected through further prompting. The episode shows the workflow's capacity for repair. It also shows how the fault was found, which was through use and not through a test written in advance. The tutorial does not report known-answer tests of retrieval from the corpus. It also notes that developers may prefer to share a technical guide in place of code that contains personal API keys. By the authors' own four conditions, an instrument shared in that way would retain prototype status, and independent verification of what it retrieves would be correspondingly limited.

On linguistic validity, the case is strong at the level of the corpus and less clear at the level of the output. The corpus was compiled and documented independently of the model. The target features were grounded in established accounts of stance and hedging and operationalised

through criteria drawn from prior corpus analysis. The tutorial states that summaries are produced from concordance hits retrieved from the corpus, so that they can be verified against it. However, an early development prompt asks that the application generate sentences similar to those in the corpus files instead of retrieving them, and the vocabulary and feedback modules produce exercises and corrections through the model. VetLingua belongs to the second kind of tool distinguished above, in which AI acts in operation. For such a tool, linguistic validation would need to establish which outputs are retrieved and which generated, whether users can tell them apart and how often generated explanations agree with expert reading of the same lines. This bears on a tension in the tutorial's guidance on records. Failed iterations are to be removed so that a clear blueprint remains, yet divergent outputs are also described as methodological data to be documented. A validation record needs the second.

On pedagogical validity, the authors are explicit that evidence does not yet exist. VetLingua had not been deployed in a classroom, pilot feedback had been used informally, and the planned evaluation combines pre- and post-intervention measures with interaction data and user outputs. The framework adds two requirements to that plan. A post-test taken with the tool available would show supported performance, whereas a delayed task with assistance reduced would show whether learners can consult evidence unaided. Because the tool supplies explanations alongside examples, logs would also need to show whether learners inspect the examples at all. None of this is a criticism of a first articulation that states its own limits. It indicates the distance between a documented prototype and the claims that such tools are likely to attract once they are in use.

Table 2. The three forms of validation applied to the published account of VetLingua

Form of validation	What the published tutorial reports	What remains to be shown
Computational correctness	Iterative prompting and debugging. A language-detection fault (English queries answered in Spanish) reported by users and corrected. A recommendation to retest earlier functionality after each prompt cycle.	Known-answer tests of corpus retrieval. Expected results fixed independently of the model. A test record that others can inspect.
Linguistic validity	An independently compiled and documented disciplinary corpus. Constructs grounded in accounts of stance and hedging and checked by the researchers. Summaries built from retrieved concordance hits.	Which outputs are retrieved and which generated, and whether users can tell. Agreement between generated explanations or feedback and expert reading.
Pedagogical validity	Three learning modules. Informal pilot feedback. Access-level analytics. A planned evaluation with pre- and post-intervention measures.	Classroom deployment. Logs showing whether learners inspect corpus evidence. Delayed and transfer tasks with assistance reduced.

Note. Based solely on Crosthwaite and Özer (2026). The instrument was not tested for this article.

Implications for design, teacher education and institutions

Interface design can make such evidence seeking more feasible. CorpusMate and ChatAI indicate ways to connect established corpus functions with more accessible interaction

(Anthony, 2025; Crosthwaite & Baisa, 2024). A vibe-coded application could adapt those functions to a local discipline or language. The design should allow users to recover the query, inspect sources and change a relevant setting. Automatic explanations should be distinguishable from retrieved examples. A learner who can see how a result was produced has a better basis for questioning it. Nevertheless, transparency itself is a design property. Whether learners use it productively remains an empirical question.

Teacher education should connect creation to judgement. A useful training task would ask teachers to describe a learning objective before requesting software. They would then compare the generated application with that objective and inspect a small set of known examples. Ma et al. (2026) provide a basis for connecting corpus literacy, pedagogical knowledge and self-efficacy. Kohnke et al. (2023) identify competencies relevant to ethical and effective chatbot use. Vibe coding requires additional preparation for specifying and evaluating instruments. That preparation can be shared across a team. Pedagogical expertise and technical review do not have to reside in the same individual.

This shared approach offers a more feasible institutional model. Research groups could maintain a small number of documented corpus components that teachers adapt with support. Institutions would provide technical review, appropriate data storage and scheduled maintenance. Local users would retain control over objectives, corpus selection and teaching tasks. The arrangement could reduce repeated development work while allowing contextual adaptation. It should also identify who responds when a tool fails. Kern (2024) connects language education to the wider conditions of technological mediation. His perspective directs attention to how institutions organise use, alongside the features of an individual application.

Research agenda

The first research priority is to compare complete development workflows. A study could assign comparable corpus tasks to conventional programming, documented AI-assisted development and exploratory vibe coding. It should examine analytical accuracy, development time and the effort required to detect and repair errors. A functioning prototype is only one outcome. Maintenance and the ability of another researcher to reproduce the analysis are also relevant. Crosthwaite and Özer (2026) provide a starting workflow for this investigation. The comparison would test when natural language development improves practical access and when its apparent speed creates additional checking costs.

A second priority is to establish reusable evaluation materials. These could include permitted sample corpora, documented expected outputs and challenging linguistic cases. They should represent different genres, languages and learner populations. Lu (2010) evaluation of automated syntactic analysis illustrates why a tool should be checked against data beyond the material used to develop it. New corpus applications also need evaluation after substantial software or model changes. An accumulated record of failures would be useful evidence. It would show where a development method needs additional support and prevent each team from discovering the same weaknesses independently.

A third priority concerns the design of human-AI work. Vaccaro et al. (2024) show through a systematic review and meta-analysis that combining humans and AI does not guarantee performance beyond the better performer alone. This evidence spans tasks outside corpus learning, so it supplies a caution rather than a direct prediction. Corpus studies should examine which decisions humans make, what information AI supplies and how disagreements are resolved. An expert checking every result may produce high accuracy at an unsustainable cost. An untrained user accepting every result may produce apparent efficiency with limited validity. The relevant unit of evaluation is the complete workflow under realistic conditions.

Pedagogical studies should also separate outcomes more carefully. They need measures of immediate performance, delayed learning and transfer to a new task. They should examine

differences associated with prior corpus experience and language proficiency. Boulton and Vyatkina (2024) call for broader methodological approaches in DDL research. Han (2024) likewise argues for systematic inquiry into ChatGPT and second language acquisition. These concerns support studies that combine outcome measures with observations of evidence use. A favourable questionnaire can indicate acceptability. It cannot establish that learners have developed language knowledge or the ability to evaluate a new claim.

Claims about replacing teachers require particular restraint. Handley (2024) questions whether AI has made language teaching obsolete. For vibe-coded corpus tools, teacher involvement remains relevant to selecting tasks, interpreting learner difficulties and adjusting support. Automation may change how that work is organised. Its educational consequences will depend on which decisions are transferred and what expertise remains available. Studies should report the teacher input required to achieve their results. Otherwise, a tool may receive credit for learning that depended on sustained human guidance. Reporting this input also makes replication and realistic adoption more feasible.

Future research should extend the populations and purposes represented in corpus-AI studies. O'Keeffe (2021) calls for a broader research gaze in DDL. Curry and McEnery (2025) identify research directions connecting corpus pedagogy to curricula, AI and wider language education. Vibe coding could help develop applications for less commonly taught languages or under-resourced settings. That possibility needs locally grounded evaluation. Researchers should involve relevant teachers and language communities in defining useful outputs. They should also investigate whether the supporting models and corpus resources handle local variation adequately. Rapid development alone cannot resolve the absence of suitable language data.

Critical literacy should become observable in this research. Pérez-Paredes et al. (2025) connect corpus work to critical engagement with AI. Building on that connection, a study could ask whether learners identify a misleading explanation, select an appropriate comparison or justify a revision using source evidence. These actions would provide stronger grounds for claims about critical learning than general confidence alone. They would also help explain differences between designs that provide similar answers. The educational contribution of a corpus-AI interface may lie in how users investigate a claim, rather than the amount of information the interface supplies.

Crosthwaite (2026) challenges the claim that generative AI makes DDL obsolete. The present argument extends that debate to the tools through which DDL and corpus research are conducted. AI-supported development could make specialised instruments more accessible. Its lasting contribution would depend on reliable components, documented decisions and support that survives the initial prototype. The strongest future trend would therefore be a growth in accountable instrument development and evidence-based learning practices. A rapid increase in custom applications would be insufficient evidence of that change. Research must establish what those applications allow people to know and learn.

Limitations

This article has several limitations. It is conceptual and reports no new data. Hence, the usefulness of the framework has yet to be shown in practice. Its literature base is purposive and a systematic search might locate studies that qualify the argument. The illustrative case rests on a single published tutorial and was not checked against the tool itself, whose features may have changed since publication. The three forms of validation are separated for clarity and real projects will meet cases in which they are hard to tell apart, as when a tokenisation choice is at once a computational and a linguistic decision. Finally, the field is moving quickly. Statements here about what current tools can and cannot do should be read as dated to 2026.

Conclusion

Vibe coding could widen participation in corpus-based language research and learning. It allows more researchers and teachers to influence the tools they use. However, it also places methodological decisions within prompts, generated code and interface defaults. Those decisions need explicit justification. The proposed framework links computational checking to linguistic interpretation and pedagogical evaluation. It explains why correct outputs, credible language claims and learning gains require different evidence.

The contribution of current corpus-AI research is to make this evaluation possible. Corpus pedagogy clarifies the role of teacher knowledge and context. Corpus literacy supports the inspection of language evidence. Methodological research connects instrument development to documentation and replication. The present framework adds the step that follows documentation: stating which claim a tool is meant to support and gathering the evidence that claim requires. These foundations should guide future studies of vibe coding. Comparative workflow studies and delayed learning measures are especially needed.

The future of this approach will depend on how institutions support it. Independent checking, shared technical expertise and continued maintenance can make local innovation more sustainable. Without that support, easier creation may expand the number of tools faster than the capacity to evaluate them. Vibe coding becomes a credible research and educational method when users can explain how their tools work, why the evidence supports their claims and what learners can do independently after using them.

Declaration of Interest Statement

There is no potential conflict of interest.

Generative AI statement

Generative AI was employed solely to enhance linguistic clarity during the final proofreading stage. All conceptualisation, analysis and interpretive judgements remain entirely the author's own.

Funding

There is no funding received.

References

Alqheedan, A., & Alzughaibi, S. (2026). The VIBE-HI framework: A conceptual model for evaluating vibe coding appropriateness, quality, and safety in health informatics. *Frontiers in Artificial Intelligence, 9*, Article 1882814. https://doi.org/10.3389/frai.2026.1882814

Anthony, L. (2025). Integrating AI technology into corpus-based language learning through ChatAI. *Computer Assisted Language Learning*. Advance online publication. https://doi.org/10.1080/09588221.2025.2589747

Baker, P., Gabrielatos, C., KhosraviNik, M., Krzyżanowski, M., McEnery, T., & Wodak, R. (2008). A useful methodological synergy? Combining critical discourse analysis and corpus linguistics to examine discourses of refugees and asylum seekers in the UK press. *Discourse & Society*, *19*(3), 273–306. https://doi.org/10.1177/0957926508088962

Barrot, J. S. (2023). Using ChatGPT for second language writing: Pitfalls and potentials. *Assessing Writing*, *57*, Article 100745. https://doi.org/10.1016/j.asw.2023.100745

Biber, D. (1993). Representativeness in corpus design. *Literary and Linguistic Computing*, *8*(4), 243–257. https://doi.org/10.1093/llc/8.4.243

Biber, D., Conrad, S., & Cortes, V. (2004). If you look at …: Lexical bundles in university teaching and textbooks. *Applied Linguistics*, *25*(3), 371–405. https://doi.org/10.1093/applin/25.3.371

Boulton, A. (2010). Data-driven learning: Taking the computer out of the equation. *Language Learning*, *60*(3), 534–572. https://doi.org/10.1111/j.1467-9922.2010.00566.x

Boulton, A., & Cobb, T. (2017). Corpus use in language learning: A meta-analysis. *Language Learning*, *67*(2), 348–393. https://doi.org/10.1111/lang.12224

Boulton, A., & Vyatkina, N. (2021). Thirty years of data-driven learning: Taking stock and charting new directions over time. *Language Learning & Technology*, *25*(3), 66–89. https://doi.org/10.64152/10125/73450

Boulton, A., & Vyatkina, N. (2024). Expanding methodological approaches in DDL research. *TESOL Quarterly*, *58*(3), 1193–1204. https://doi.org/10.1002/tesq.3269

Celik, I. (2023). Towards Intelligent-TPACK: An empirical study on teachers' professional knowledge to ethically integrate artificial intelligence (AI)-based tools into education. *Computers in Human Behavior*, *138*, Article 107468. https://doi.org/10.1016/j.chb.2022.107468

Chambers, A. (2019). Towards the corpus revolution? Bridging the research–practice gap. *Language Teaching*, *52*(4), 460–475. https://doi.org/10.1017/S0261444819000089

Charles, M. (2024). Benefits and challenges of using do-it-yourself corpora for academic writing development. *TESOL Quarterly*, *58*(3), 1205–1214. https://doi.org/10.1002/tesq.3276

Cheung, L., & Crosthwaite, P. (2025). CorpusChat: Integrating corpus linguistics and generative AI for academic writing development. *Computer Assisted Language Learning*. Advance online publication. https://doi.org/10.1080/09588221.2025.2506480

Crosthwaite, P. (2026). Is DDL dead yet? Not so fast! Revisiting earlier claims on the state of corpus linguistics, DDL, generative AI, and Language for Specific Purposes. *Ibérica*, *51*, 5–19. https://doi.org/10.17398/2340-2784.51.5

Crosthwaite, P., & Baisa, V. (2023). Generative AI and the end of corpus-assisted data-driven learning? Not so fast! *Applied Corpus Linguistics*, *3*(3), Article 100066. https://doi.org/10.1016/j.acorp.2023.100066

Crosthwaite, P., & Baisa, V. (2024). A user-friendly corpus tool for disciplinary data-driven learning: Introducing CorpusMate. *International Journal of Corpus Linguistics*, *29*(4), 595–610. https://doi.org/10.1075/ijcl.23056.cro

Crosthwaite, P., & Özer, M. (2026). Vibecoding as method: A structured workflow for AI-assisted research instrument development in applied linguistics. *Research Methods in Applied Linguistics*, *5*(3), Article 100356. https://doi.org/10.1016/j.rmal.2026.100356

Crosthwaite, P., Sun, S. A., & Jiang, F. K. (2026). Prompt sensitivity in AI writing: A methodological caution for corpus-based analysis of AI-generated language. *The Modern Language Journal*. Advance online publication. https://doi.org/10.1111/modl.70089

Curry, N., & McEnery, T. (2025). Corpus linguistics for language teaching and learning: A research agenda. *Language Teaching*, *58*(2), 232–251. https://doi.org/10.1017/S0261444824000430

Dong, J., & Wang, H. (2025). Exploring the effectiveness of a DDL-GenAI teaching mode in students' EFL writing classroom. *Computer Assisted Language Learning*. Advance online publication. https://doi.org/10.1080/09588221.2025.2577369

Elgendy, I. A., Dwivedi, Y. K., Al-Sharafi, M. A., Hosny, M., Helal, M. Y., Crick, T., & Al-Busaidi, A. S. (2026). Responsible vibe coding: Architecture, opportunities, and research agenda. *Journal of Computer Information Systems*. Advance online publication. https://doi.org/10.1080/08874417.2026.2621186

Frankenberg-Garcia, A. (2014). The use of corpus examples for language comprehension and production. *ReCALL*, *26*(2), 128–146. https://doi.org/10.1017/S0958344014000093

Gries, S. T. (2008). Dispersions and adjusted frequencies in corpora. *International Journal of Corpus Linguistics*, *13*(4), 403–437. https://doi.org/10.1075/ijcl.13.4.02gri

Han, Z. (2024). ChatGPT in and for second language acquisition: A call for systematic research. *Studies in Second Language Acquisition*, *46*(2), 301–306.

https://doi.org/10.1017/S0272263124000111
Handley, Z. (2024). Has artificial intelligence rendered language teaching obsolete? *The Modern Language Journal*, *108*(2), 548–555. https://doi.org/10.1111/modl.12929
Kane, M. T. (2013). Validating the interpretations and uses of test scores. *Journal of Educational Measurement*, *50*(1), 1–73. https://doi.org/10.1111/jedm.12000
Karpathy, A. [@karpathy]. (2025, February 2). *There's a new kind of coding I call "vibe coding"* [Post]. X. https://x.com/karpathy/status/1886192184808149383
Kern, R. (2021). Twenty-five years of digital literacies in CALL. *Language Learning & Technology*, *25*(3), 132–150. https://doi.org/10.64152/10125/73453
Kern, R. (2024). Twenty-first century technologies and language education: Charting a path forward. *The Modern Language Journal*, *108*(2), 515–533. https://doi.org/10.1111/modl.12924
Kilgarriff, A. (2005). Language is never, ever, ever, random. *Corpus Linguistics and Linguistic Theory*, *1*(2), 263–276. https://doi.org/10.1515/cllt.2005.1.2.263
Kohnke, L., Moorhouse, B. L., & Zou, D. (2023). ChatGPT for language teaching and learning. *RELC Journal*, *54*(2), 537–550. https://doi.org/10.1177/00336882231162868
Lu, X. (2010). Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, *15*(4), 474–496. https://doi.org/10.1075/ijcl.15.4.02lu
Lu, X. (2012). The relationship of lexical richness to the quality of ESL learners' oral narratives. *The Modern Language Journal*, *96*(2), 190–208. https://doi.org/10.1111/j.1540-4781.2011.01232.x
Ma, Q., Yan, J., Crosthwaite, P., & Lee, C.-K. J. (2026). TPACK for corpus technology: TESOL student teachers' growth and self-efficacy in independent language learning and teaching. *ReCALL*, *38*(1), 94–110. https://doi.org/10.1017/S0958344025100347
Ma, Q., Yuan, R. E., Cheung, L. M. E., & Yang, J. (2024). Teacher paths for developing corpus-based language pedagogy: A case study. *Computer Assisted Language Learning*, *37*(3), 461–492. https://doi.org/10.1080/09588221.2022.2040537
Marsden, E., Morgan-Short, K., Thompson, S., & Abugaber, D. (2018). Replication in second language research: Narrative and systematic reviews and recommendations for the field. *Language Learning*, *68*(2), 321–391. https://doi.org/10.1111/lang.12286
McEnery, T., & Hardie, A. (2012). *Corpus linguistics: Method, theory and practice*. Cambridge University Press.
Messeri, L., & Crockett, M. J. (2024). Artificial intelligence and illusions of understanding in scientific research. *Nature*, *627*, 49–58. https://doi.org/10.1038/s41586-024-07146-0
Mizumoto, A. (2023). Data-driven learning meets generative AI: Introducing the framework of metacognitive resource use. *Applied Corpus Linguistics*, *3*(3), Article 100074. https://doi.org/10.1016/j.acorp.2023.100074
Mizumoto, A., & Eguchi, M. (2023). Exploring the potential of using an AI language model for automated essay scoring. *Research Methods in Applied Linguistics*, *2*(2), Article 100050. https://doi.org/10.1016/j.rmal.2023.100050
Ng, D. T. K., Leung, J. K. L., Chu, S. K. W., & Qiao, M. S. (2021). Conceptualizing AI literacy: An exploratory review. *Computers and Education: Artificial Intelligence*, *2*, Article 100041. https://doi.org/10.1016/j.caeai.2021.100041
O'Keeffe, A. (2021). Data-driven learning – a call for a broader research gaze. *Language Teaching*, *54*(2), 259–272. https://doi.org/10.1017/S0261444820000245
O'Sullivan, Í. (2007). Enhancing a process-oriented approach to literacy and language learning: The role of corpus consultation literacy. *ReCALL*, *19*(3), 269–286. https://doi.org/10.1017/S095834400700033X
Ouyang, S., Zhang, J. M., Harman, M., & Wang, M. (2025). An empirical study of the non-determinism of ChatGPT in code generation. *ACM Transactions on Software Engineering and Methodology, 34*(2), 1–28. https://doi.org/10.1145/3697010
Pérez-Paredes, P. (2022). A systematic review of the uses and spread of corpora and data-driven learning in CALL research during 2011–2015. *Computer Assisted Language*

Learning, *35*(1–2), 36–61. https://doi.org/10.1080/09588221.2019.1667832

Pérez-Paredes, P. (2026). Corpus linguistics and AI in the reconfiguration of language learning ecologies. *Language Teaching*. Advance online publication. https://doi.org/10.1017/S0261444825101109

Pérez-Paredes, P., Curry, N., & Aguado Jiménez, P. (2025). Integrating critical corpus and AI literacies in applied linguistics: A mixed-methods study. *Computer Assisted Language Learning*. Advance online publication. https://doi.org/10.1080/09588221.2025.2569351

Pérez-Paredes, P., Sánchez-Tornel, M., & Alcaraz Calero, J. M. (2012). Learners' search patterns during corpus-based focus-on-form activities: A study on hands-on concordancing. *International Journal of Corpus Linguistics*, *17*(4), 482–515. https://doi.org/10.1075/ijcl.17.4.02par

Pérez-Paredes, P., Sánchez-Tornel, M., Alcaraz Calero, J. M., & Aguado Jiménez, P. (2011). Tracking learners' actual uses of corpora: Guided vs non-guided corpus consultation. *Computer Assisted Language Learning*, *24*(3), 233–253. https://doi.org/10.1080/09588221.2010.539978

Ren, Y., Wong, W. L., Barrot, J., & Zhang, L. J. (2026). Learning styles, engagement and anxiety in AI-mediated writing: A multimodal feedback study. *International Journal of Applied Linguistics*. Advance online publication. https://doi.org/10.1111/ijal.70120

Risko, E. F., & Gilbert, S. J. (2016). Cognitive offloading. *Trends in Cognitive Sciences*, *20*(9), 676–688. https://doi.org/10.1016/j.tics.2016.07.002

Shumailov, I., Shumaylov, Z., Zhao, Y., Papernot, N., Anderson, R., & Gal, Y. (2024). AI models collapse when trained on recursively generated data. *Nature*, *631*, 755–759. https://doi.org/10.1038/s41586-024-07566-y

Sinclair, J. (2005). Corpus and text: Basic principles. In M. Wynne (Ed.), *Developing linguistic corpora: A guide to good practice* (pp. 1–16). Oxbow Books.

Vaccaro, M., Almaatouq, A., & Malone, T. W. (2024). When combinations of humans and AI are useful: A systematic review and meta-analysis. *Nature Human Behaviour*, *8*, 2293–2303. https://doi.org/10.1038/s41562-024-02024-1

Wang, Q., Deng, Y., & Liu, D. (2025). Corpus-driven and AI-assisted learning in teaching business academic writing: Examining their efficacies and the predicting/moderating effects of learner variables. *Computer Assisted Language Learning*. Advance online publication. https://doi.org/10.1080/09588221.2025.2589739

Wong, W. L., & Fong, S. (2025). A new era with artificial intelligence powered learner corpora for teaching and learning of English as a second language in primary school education. *Applied Language Sciences, 1*, Article 250105. https://doi.org/10.65553/ALS.250105

Wong, W. L., Abuhassna, H., Barrot, J., & Zhang, L. J. (2026). Trends in corpus linguistics and artificial intelligence language technology research: A bibliometric review. *Asian-Pacific Journal of Second and Foreign Language Education*, *11*, Article 35. https://doi.org/10.1186/s40862-026-00412-w

Zhang, N., Wong, W. L., & Barrot, J. (2025). Primary teachers' acceptance and sustained adoption of AI powered learner corpora for writing instruction through TAM and ECM perspectives. *Scientific Reports, 15*, Article 37399. https://doi.org/10.1038/s41598-025-21205-0

Author Bios

Wei Lun Wong is a senior lecturer at Faculty of Education, Universiti Kebangsaan Malaysia. His research interests include GenAI learner corpora, teacher education and applied linguistics. His works appear at *International Journal of Applied Linguistics, The Asia-Pacific Education Research, Educational Review, SAGE Open* and *Scientific Reports*.