Available on: https://jsshrs.ep-journals.org/ E-ISSN: 3051-3561, P-ISSN: 3051-3553

Journal of Social Science and Human Research Studies

Volume : 02 Issue : 10 October-2026

Arabic Compatible Word Analysers for Corpus Research: An Affordance Based Critical Review and Evaluation Framework

Zainur Rijal Abdul Razak
Fakulti Pengajian Bahasa Utama, Universiti Sains Islam Malaysia, 71800 Nilai, Negeri Sembilan, Malaysia

Rosni Samah
Fakulti Pengajian Bahasa Utama, Universiti Sains Islam Malaysia, 71800 Nilai, Negeri Sembilan, Malaysia

Published Date: October 19, 2026 **Article DOI: 10.65150/EP-jsshrs/V2E10/2026-08**

ABSTRACT: Arabic corpus researchers need software that can transform written text into linguistically interpretable evidence. However, displaying Arabic characters, retrieving orthographic words and identifying morphological units are different capabilities, and their conflation can obscure the conditions under which corpus findings remain defensible. This article develops an affordance based critical review of Arabic compatible word analysers through a synthesis of published software studies and current developer documentation. It examines AntConc, WordSmith Tools, LancsBox, Sketch Engine, aConCorde, the Buckwalter Arabic Morphological Analyzer, the Standard Arabic Morphological Analyzer, MADAMIRA and CAMeL Tools. The review draws selectively on developer documentation, resource catalogues and foundational publications rather than reporting an experimental benchmark or systematic review. Its central problem concerns the relationship between documented software capabilities and valid analytical action in Arabic corpus research. The proposed framework distinguishes script access, linguistic representation, contextual retrieval, statistical comparison and reproducibility, with researcher competence and corpus variety treated as conditions of affordance actualisation. The analysis indicates that concordancers and morphological processors offer complementary capabilities rather than interchangeable solutions. It proposes a layered workflow and a controlled evaluation design through which future studies can test retrieval accuracy, segmentation decisions and the stability of corpus interpretations. The contribution is a theoretically grounded basis for software selection and evaluation that connects technical functions to research questions without assuming that tool availability establishes analytical validity.

KEYWORDS: Arabic corpus linguistics, word analysers, affordance theory, concordance, morphological analysis, reproducibility.

1. INTRODUCTION

Computer assisted corpus analysis makes it possible to investigate patterns across texts that exceed the practical limits of manual reading. Word lists organise recurring forms, concordances return expressions to their contexts, and keyword procedures compare their distributions across collections. These functions provide a starting point for Arabic linguistic research, including investigations of news discourse, academic vocabulary, learner language and Islamic texts. Nevertheless, a program can produce consistent numerical output while its counting units remain poorly aligned with the research question. The methodological issue therefore extends beyond whether Arabic characters can be imported successfully. Arabic corpus software encompasses two related areas of practice which are exploring distributions across texts and preparing linguistic representations for analysis. Concordancers typically foreground frequency, contextual retrieval and comparison, whereas morphological processors identify or select analyses of written forms. Their functions overlap within research workflows, but their outputs are not interchangeable. Evaluating these systems together therefore requires criteria that recognise both their distinct purposes and their possible integration. The central concern is how researchers can select and combine tools while maintaining a defensible connection between linguistic questions, counted units and contextual interpretation. Tool selection should therefore begin with the specific intended linguistic claim and evidence needed to substantiate that claim.

Arabic makes these distinctions consequential. Written forms can contain attached conjunctions, prepositions, articles and pronouns. Short vowels are often absent from ordinary writing, while inflection and derivation produce relationships that a character-based search cannot independently resolve. A frequency list of written forms therefore answers a different question from a frequency list of lemmas or roots. Habash (2010) provides the linguistic and computational background for these representational problems. Their importance here lies in the connection between representation, and the claims researchers subsequently make. Existing publications already discuss Arabic morphology, Arabic oriented concordancing and comparative software evaluation (Alfaifi & Atwell, 2016). Al-Sughaiyer and Al-Kharashi (2004) survey morphological techniques, while Roberts et al. (2006) describe aConCorde. Consequently, the research problem cannot responsibly be framed as the absence of Arabic software or the absence of all previous evaluation. A more defensible problem concerns how capabilities distributed across software categories should be evaluated together when researchers move from Arabic text to corpus interpretation. The proposed extension is to connect

License: This is an open access article under the CC BY 4.0 license: https://creativecommons.org/licenses/by/4.0/

earlier compatibility evaluations with researcher actions, morphological representation and interpretive stability; an exhaustive novelty claim would require broader literature verification.

The research problem concerns the conditions under which Arabic compatible word analysers support accurate, interpretable and reproducible corpus investigation across different tasks and textual varieties. Affordance theory provides an explanatory framework because the question concerns relationships among tools, users, texts and purposes. A feature inventory alone cannot explain why the same function helps one researcher investigate a question while leaving another researcher unable to retrieve or interpret the relevant evidence. Three questions organise the article. First, which documented capabilities support Arabic script access, linguistic representation and contextual investigation? Second, what advantages and constraints emerge when corpus platforms are compared with morphological processors? Third, how can affordance theory guide an evaluation protocol that links software functions to defensible research outcomes? The contribution is a critical review and a testable framework, rather than an experimentally established ranking.

2. AFFORDANCE THEORY AND ARABIC ANALYTICAL UNITS

2.1 From available functions to analytical action

Hutchby (2001) applies affordance reasoning to technologies, recognising that material properties enable and constrain action, while van Lier (2004) develops an ecological perspective relevant to language education. Together, these perspectives provide a basis for analysing corpus software through the actions it makes possible for particular users under particular conditions (Hutchby, 2001; van Lier, 2004). The present framework extends this relational reasoning to corpus research practice; it does not attribute an Arabic software evaluation model to these authors. Retrieving relevant Arabic occurrences with sufficient contextual information constitutes an analytical action whose successful execution depends on readable corpus input, an appropriately formulated query, indexing rules that preserve relevant distinctions, and the researcher's ability to interpret the results. Previous Arabic software evaluations demonstrate the importance of distinguishing text import from readable contextual presentation and normalization support (Alfaifi & Atwell, 2016). Similarly, Arabic processing research distinguishes preprocessing, morphological analysis and contextual disambiguation as separate operations (Obeid et al., 2020; Pasha et al., 2014). These distinctions prevent *affordance* from becoming merely a synonym for technical capability and make researcher knowledge analytically relevant: familiarity with query syntax and morphological annotation can influence which analytical possibilities a user can recognise and actualise.

The present article proposes five affordance domains to operationalise the framework, first, script access concerns preservation and readable presentation, second, linguistic representation concerns the units available for indexing, third, contextual retrieval concerns the recovery and interpretation of occurrences, fourth, statistical comparison concerns distributions calculated over compatible units, and fifth, reproducibility concerns the ability to reconstruct the workflow. These domains synthesise issues addressed separately in Arabic software evaluation, morphological processing and corpus statistics rather than reproduce an established five-domain classification (Alfaifi & Atwell, 2016; Obeid et al., 2020). Although the domains overlap, distinguishing them enables reviewers to identify the evidence needed to substantiate each claim.

Within this framework, visual accessibility and linguistic adequacy require separate assessment: readable contextual presentation does not establish that tokenization or annotation corresponds to the research question. Morphological processing requires explicit representational decisions, while statistical analysis requires procedures appropriate to the data and intended comparison (Brezina, 2018; Maamouri et al., 2010). Affordance theory therefore explains the relationship between available capabilities and possible analytical actions, but decisions about lemmas, statistical thresholds and annotation accuracy require additional linguistic and methodological evidence. An operational evaluation must pair theoretical categories with observable actions, explicit success criteria and validation procedures; otherwise, it risks reproducing unsupported judgments about software advantages.

2.2 Orthographic words and morphological representations

Arabic corpus research must specify what counts as a word because orthographic tokenization and morphological segmentation establish different analytical units (Obeid et al., 2020). An orthographic token is defined through written boundaries and tokenization rules, whereas morphological segmentation may distinguish attached material from a lexical base. Arabic processing documentation explicitly provides different segmentation schemes, demonstrating that the resulting units depend on the selected convention (CAMeL Lab, n.d.). A lemma groups forms according to lexical identity, whereas a root groups them according to a different morphological relationship (Habash, 2010). These representations should therefore be evaluated according to the research question rather than treated as successive levels of automatic improvement.

For an investigation of written variation, retaining orthographic distinctions may be essential; for lexical distribution, grouping forms under verified lemmas may be appropriate, and for pronoun attachment, clitic segmentation can make the target available for analysis. These applications follow from the distinctions among surface forms, morpheme boundaries and lexical analyses documented in Arabic morphological resources (Buckwalter, 2004; Maamouri et al., 2010). Root-based grouping can support investigation of derivational families, but shared roots should not be assumed to establish identical meanings or grammatical functions (Habash, 2010). The methodological principle is therefore alignment: researchers should select a representation corresponding to their analytical construct while preserving access to the original text.

Diacritics introduce another representational decision because their omission contributes to ambiguity in Arabic processing (Habash, 2010; Obeid et al., 2020). Removing them may facilitate retrieval across differently vocalised forms, but it also removes information that may be relevant to morphological interpretation. Similarly, normalization operations can merge letter variants, making their suitability dependent on whether the research concerns lexical retrieval or orthographic distinctions (CAMeL Lab, n.d.-b). The Unicode Consortium (2024) distinguishes character representation from Arabic shaping and bidirectional display. These provide technical foundations for reliable text handling, but successful rendering does not establish correct segmentation, lemma assignment or contextual disambiguation (Maamouri et al., 2010; Unicode Consortium, 2024). Consequently, a readable screen cannot by itself validate the linguistic representation used in the index.

License: This is an open access article under the CC BY 4.0 license: https://creativecommons.org/licenses/by/4.0/

METHODOLOGY

This article adopts a critical literature review to synthesise previous research and evaluate the analytical suitability of Arabic-compatible corpus software. A critical review moves beyond describing individual publications by examining their contributions and developing an interpretive argument or conceptual framework (Grant & Booth, 2009). This approach suits the article's objective of connecting software capabilities, reported advantages and disadvantages, and the conditions under which researchers can use these capabilities effectively. The review combines published evaluations, software development studies and official documentation to construct an affordance-based comparison. Software selection was purposive, encompassing desktop concordancers, integrated corpus platforms and Arabic morphological processors. Inclusion prioritised relevance to Arabic corpus investigation, identifiable scholarly or official sources, and sufficient information to establish the software's analytical role. The selected tools represent complementary activities, including frequency analysis, concordancing, keyword comparison, morphological segmentation and disambiguation. Selection therefore supports comparison across research tasks rather than an exhaustive inventory of Arabic applications. Consistent with Snyder (2019), the review approach and source selection were aligned with the research questions and intended conceptual contribution.

English search terms combined software names with *Arabic*, *Unicode*, *tokenization*, *morphology*, *concordance* and *affordances*. Sources included the ACL Anthology, Linguistic Data Consortium catalogues, publisher websites and official software documentation. Published comparative studies provided evidence about previously evaluated configurations, while developer documentation clarified available functions and processing options. The search was selective rather than exhaustive; consequently, the article is presented as a critical review rather than a systematic review. The analysis used a structured comparison of each tool's principal purpose, documented capabilities, advantages, limitations and potential research applications. These characteristics were interpreted through five domains proposed in this article: script access, linguistic representation, contextual retrieval, statistical comparison and reproducibility. The domains served as an organising framework for synthesis rather than a validated scoring instrument. Affordance theory informed the interpretation of how software properties enable or constrain analytical actions under particular user and corpus conditions (Hutchby, 2001; van Lier, 2004).

Evidence appraisal distinguished published evaluation findings from developer descriptions and the present article's methodological interpretations. Findings were retained within the boundaries of the evaluated release, configuration, corpus and task. Historical limitations were therefore not assumed to persist in current versions, while documented functionality was not treated as proof of accuracy on an untested corpus. The synthesis identifies complementary capabilities and unresolved evaluation needs without producing a universal ranking, pooled performance estimate or new experimental results. This separation of evidence and interpretation supports the critical review's conceptual contribution while preserving its methodological limits.

4. CRITICAL REVIEW OF ARABIC COMPATIBLE SOFTWARE

4.1 AntConc

AntConc is a freeware desktop corpus toolkit offering concordancing and lexical analysis. Its official documentation identifies UTF-8 as the default encoding and provides configurable token definitions; its interface includes word lists, keywords, clusters, n-grams and collocates (Anthony, 2026). These functions make it a relevant starting point for inspecting researcher assembled Arabic texts. Its principal advantage, assessed here from documented functionality, is accessible local exploration without requiring a complete programming pipeline. Researchers can examine orthographic distributions and return selected forms to their contexts. These are documented capabilities rather than findings from an Arabic benchmark. Its limitations follow from the distinction between character-based access and contextual morphological interpretation. Importing Arabic does not establish automatic Arabic lemma assignment or reliable clitic disambiguation. A researcher may need prepared annotations, grouping resources or externally processed data, depending on the question. Diacritic handling and token definitions also require inspection. AntConc is therefore a reasonable candidate for exploratory analysis and teaching demonstrations, if users validate Arabic indexing and presentation on their own material before interpreting output as lexical evidence.

4.2 WordSmith Tools

WordSmith Tools supports lexical investigation through its WordList, Concord and KeyWords components. Official version 9 documentation discusses Unicode and language settings, including Arabic; the developer describes the software as a Windows application (Scott, n.d.). Its advantage is an established environment for local lexical investigation and corpus comparison. Researchers familiar with its workflow can retain an explicit connection among frequency lists, contextual evidence and comparative statistics. Using the same environment across stages may also reduce unnecessary changes in export formats. The principal constraints are platform dependence, licensing and the need to configure the linguistic units appropriately. Unicode access is necessary but does not establish context sensitive Arabic morphological analysis. Researchers working across several operating systems must consider whether all collaborators can reproduce the setup. They must also document the exact release and settings rather than transfer assumptions from earlier WordSmith studies. Its suitability is strongest when local corpus control and an established analysis workflow matter, with external Arabic processing added where the research question requires it.

4.3 LancsBox and LancsBox X

LancsBox supports corpus exploration through concordances, lexical summaries and collocation analysis. The current LancsBox X site identifies free availability for non-commercial use and includes GraphColl among its functions (Brezina & Platt, 2026). Brezina et al. (2015) explain the methodological basis for collocation networks. Its distinctive advantage is the possibility of examining relationships beyond a single frequency list, allowing researchers to connect contextual inspection with network representations. Such visualisation can generate questions about distributional associations that deserve closer linguistic examination. The constraint is that an interpretable network depends on valid input units and declared statistical settings. A graph does not independently correct segmentation or establish semantic relations. Current Arabic tokenization, annotation and bidirectional presentation should be tested in the selected release rather than inferred from general multilingual functionality or older descriptions. LancsBox is a promising candidate for collocation-oriented Arabic projects, but this review makes no verified claim about built-in Arabic morphological accuracy. Its strongest proposed role is exploratory relationship analysis followed by concordance checking.

License: This is an open access article under the CC BY 4.0 license: https://creativecommons.org/licenses/by/4.0/

4.4 Sketch Engine

Sketch Engine explicitly documents Arabic concordances, word lists, n-grams, keyword or term analysis and word sketches. Its Arabic resources include corpora from different domains, while linguistic annotation varies by corpus and processing configuration (Lexical Computing, n.d.). Its advantage is an integrated environment linking corpus access and several analytical functions. Researchers can investigate prepared resources and build user corpora, subject to access terms. This integration can support projects requiring contextual retrieval alongside grammatically organised summaries and reference corpus comparisons. Constraints include subscription or institutional access, dependence on the chosen annotation pipeline and the need to record corpus identity. An Arabic word sketch reflects both language use and the annotation and grammar underlying its construction. Researchers should not generalise one corpus configuration to every Arabic resource in the platform. Sketch Engine is consequently a useful candidate for projects benefiting from integrated annotated corpora, provided that corpus composition, processing decisions and comparison settings remain visible. Platform convenience does not eliminate responsibility for checking the evidence behind an interpretation.

4.5 aConCorde

Roberts et al. (2006) present aConCorde as an Arabic oriented concordancer intended to address problems encountered with more general tools. Its significance is methodological as well as historical because Arabic contextual retrieval involves presentation and search requirements that cannot be reduced to character import. Its design directs attention to the relationship between language specific interface requirements and the interpretability of retrieved evidence. An Arabic focused design offers a useful example of aligning software development with linguistic and instructional needs. However, a historical publication cannot establish contemporary maintainability. Current distribution, installation, dependency support and reproducible operation require separate checks. Its inclusion therefore supports discussion of Arabic centred design rather than an unconditional recommendation for new projects. Researchers evaluating aConCorde should distinguish the published concept from the executable release they can obtain. Its enduring contribution to this comparison is the recognition that usable Arabic concordancing requires attention to how users retrieve, read and interpret contexts, not simply whether strings are accepted.

4.6 Buckwalter Arabic Morphological Analyzer

The Buckwalter Arabic Morphological Analyzer version 2.0 generates possible prefix, stem and suffix analyses with linguistic labels and glosses (Buckwalter, 2004). It belongs to a different category from AntConc or WordSmith where its central purpose is morphological representation rather than interactive corpus exploration. Its advantage is explicit access to possible internal structures of written forms. This can help researchers investigate ambiguity and prepare annotation resources, especially where an orthographic word contains material relevant to more than one analytical unit. Its key limitation is that possible analyses are not equivalent to a contextually selected interpretation. Counting every candidate would confuse alternatives with observed occurrences. Additional selection, manual adjudication or a disambiguation process is required before producing a single lemma distribution. Licensing and implementation requirements also affect practical access. Buckwalter is therefore better considered a morphological component within a workflow than a complete replacement for corpus software. Its transparency is useful when ambiguity itself is relevant and researchers need to inspect the alternatives a system proposes.

4.7 Standard Arabic Morphological Analyzer (SAMA)

The Standard Arabic Morphological Analyzer (SAMA) version 3.1 extends the Buckwalter lineage. The LDC catalogue documents change in lexical resources and processing options, including UTF-8 support, while retaining the generation of possible analyses (Maamouri et al., 2010). Its advantage is a documented continuation of a recognisable morphological architecture. For researchers comparing resource versions, this continuity allows questions about how changes in lexicons and annotation inventories influence coverage without pretending that the system is an unrelated corpus platform. SAMA retains the challenge of choosing among alternatives. A larger resource may provide more candidate analyses without resolving which one fits a passage. Researchers must therefore distinguish coverage from disambiguation and inspect outputs before aggregation. Installation dependencies, resource licences and the export transformation into a searchable corpus also need documentation. Its appropriate role is morphology focused preparation and analysis in projects equipped to manage these stages. A useful comparison would assess how version changes affect particular corpus tasks rather than presume that greater lexical coverage guarantees better interpretations.

4.8 Madamira

MADAMIRA combines morphological analysis with disambiguation, drawing on earlier Arabic processing systems (Pasha et al., 2014). Its distinguishing advantage is support for choosing an analysis in context rather than presenting only an unranked inventory of possibilities. This is relevant when researchers need a processed representation for lemma queries, grammatical categories or segmented corpus exploration. Its system publication provides a foundation for understanding the architecture and intended processing tasks, although it does not validate every possible application or text domain. Its constraints include setup, resource access and transfer beyond the varieties and datasets for which the selected configuration was developed. Historical performance results should not be repeated as accuracy estimates for a new Classical Arabic, learner or dialect corpus. Errors can also become less visible after a single annotation is selected. MADAMIRA is a relevant candidate for contextual preprocessing, but researchers should retain original forms, audit samples and record the configuration. The corpus interpretation remains the researcher's responsibility even when the processor outputs a confident analysis.

4.9 CAMeL Tools

CAMeL Tools is an open-source Python toolkit with Arabic preprocessing and morphological capabilities, alongside other NLP components (Obeid et al., 2020). Developer documentation exposes configurable normalization operations and morphological tokenization schemes (CAMeL Lab, n.d.). Its advantage is workflow control because researchers can record processing operations in scripts and explicitly choose resources and representations. This makes it especially relevant to studies evaluating how decisions about text preparation affect downstream evidence. Open-source availability also enables examination of implementation details rather than reliance only on interface descriptions. The corresponding limitation is the technical competence needed to manage code, models, dependencies and data exchange. Installing the package alone does not establish that every relevant model is available or suitable for the corpus. Its annotation output also needs a searching and statistical environment if researchers want an interactive corpus workflow. CAMeL Tools is therefore a strong candidate for transparent preprocessing within a broader

License: This is an open access article under the CC BY 4.0 license: https://creativecommons.org/licenses/by/4.0/

analytical pipeline. Its flexibility must be disciplined through explicit configuration and validation; otherwise, additional options can create more undocumented variation between researchers.

4.1 Comparative overview

Table 1 summarises analytical roles rather than performance rankings. Each limitation identifies what a researcher would need to resolve before relying on the tool for the specified task.

Table 1:

Tool	Main role	Principal advantage	Principal constraint	Proposed research use
AntConc	Desktop corpus exploration	Accessible local lexical inspection	Arabic representation requires validation	Exploratory word and context analysis
WordSmith Tools	Desktop lexical comparison	Integrated wordlist and keyword workflow	Windows and licensing requirements	Controlled local corpus comparisons
LancsBox X	Corpus exploration and networks	Collocation visualisation	Arabic pipeline and graph settings require checks	Distributional relationship exploration
Sketch Engine	Integrated corpus platform	Arabic resources and annotated querying	Access terms and corpus specific annotation	Annotated corpus investigation
aConCorde	Arabic concordancing	Arabic centred design	Current executable status unverified	Historical design comparison
Buckwalter	Morphological candidate generation	Explicit possible analyses	Contextual selection needed	Ambiguity and annotation preparation
SAMA	Morphological candidate generation	Documented resource continuation	Alternatives still require selection	Morphological workflow development
MADAMIRA	Analysis and disambiguation	Contextual annotation	Domain transfer requires validation	Preprocessing for linguistic querying
CAMeL Tools	Programmable Arabic processing	Configurable scripted workflow	Coding and model management	Reproducible preprocessing

4.12 Synthesis of advantages and disadvantages from previous studies

Alfaifi and Atwell (2016) evaluated seven tools against eight Arabic support criteria. Their historical results provide direct comparative evidence:

Evaluated configuration	Reported advantage	Reported disadvantage
AntConc 3.4.0	Read Arabic files and allowed personal corpora	Reversed concordance context columns; lacked normalization
WordSmith 6.0	Read Arabic files and displayed right to left text	Misplaced diacritics; lacked normalization
aConCorde 0.4.3	Arabic interface and readable Arabic contexts	No diacritic or hamza normalization
Sketch Engine configuration tested	Arabic processing with normalization and corpus uploads	No Arabic interface at evaluation time
Khawas 3.0	Arabic interface and normalization options	UTF-16 import failed; Windows display omissions

The study's criteria mainly concern access, display and retrieval but they do not establish morphological accuracy or the validity of downstream interpretations. Khawas is included here to preserve an important comparator from that study, with current availability unverified. Across the morphological literature, a second contrast emerges. Buckwalter (2004) and Maamouri et al. (2010) document alternative generation, whereas Pasha et al. (2014) describe contextual disambiguation. Alternative generation makes ambiguity inspectable but requires selection before aggregation. Disambiguation simplifies downstream querying but can conceal uncertainty when only the selected label is retained. These advantages therefore involve different research actions rather than a universal hierarchy of quality. Obeid et al. (2020) add a toolkit perspective in which several processing components can be combined and the analytical benefit depends on documenting that combination.

The studies also have different evidential purposes. A comparative compatibility evaluation supports statements about the tested interface and retrieval conditions. A system development paper supports statements about architecture and reports performance within its own evaluation design. A methodological collocation paper supports interpretation of network procedures. Treating all three as interchangeable evidence would overstate what previous research establishes. A synthesis should instead connect each reported strength or weakness to the task examined and specify whether transfer to a new corpus has been demonstrated. The resulting research problem is more precise than asking which program is best. It asks how documented strengths become usable analytical actions, and whether those actions preserve the distinctions required by an Arabic research question. A normalization function can improve retrieval coverage while making orthographic investigation harder. An Arabic interface

License: This is an open access article under the CC BY 4.0 license: https://creativecommons.org/licenses/by/4.0/

can improve access without ensuring valid lemma counts. Automated annotation can enable large scale querying while introducing errors that require contextual checks. These are theoretical consequences to test, rather than additional findings attributed to the historical evaluation.

5. CROSS TOOL ANALYSIS OF AFFORDANCES

5.1 Word lists, Concordances and Keywords

Word list evaluation requires an important distinction between accuracy and agreement. Two programs may disagree because they apply different defensible tokenization conventions, while two programs may agree because both share an unsuitable preprocessing decision. Numerical agreement is therefore insufficient without an explicit reference representation. Researchers should report totals at each relevant level and identify which level supports their conclusions. A well specified disagreement can be more informative than an unexplained match because it reveals assumptions that would otherwise remain hidden inside software defaults. Concordances afford movement between distribution and interpretation. They allow researchers to inspect whether retrieved occurrences instantiate the intended phenomenon, whether an annotation fits the context and whether an apparent pattern depends on a small number of documents. For Arabic, evaluation must include readable bidirectional presentation and reliable links to wider passages. A short window may be sufficient for spelling inspection but inadequate for resolving a grammatical or discourse question. Context requirements should therefore be determined by the analytical task.

Keyword analysis requires a reference corpus, an identified comparison unit and an interpretable statistical procedure. A keyword list does not independently establish thematic importance. If two corpora use different segmentation or normalization, observed contrasts may partly reflect processing decisions. Researchers should align representation before comparison and examine whether prominent items occur across documents or cluster in a single source. Brezina (2018) provides methodological grounding for selecting statistics in relation to research questions. The present implication is that comparison settings belong in the evidential record. Collocation likewise depends on the units and boundaries used to calculate proximity. Splitting attached elements changes the sequence in which neighbouring tokens are counted. Excluding punctuation or crossing sentence boundaries can also change relationships. Network displays add interpretive accessibility, but their structure reflects declared thresholds and association measures. A proposed Arabic evaluation should hold these decisions constant wherever possible, then vary them deliberately to examine sensitivity. Stability across reasonable settings would strengthen an interpretation; instability would identify a need for narrower claims or further contextual examination.

5.2 Reproducibility and practical accessibility

Reproducibility requires more than naming software. Another researcher needs the corpus version, preprocessing rules, model identity, tokenization convention, query and statistical settings. For commercial platforms, access conditions may limit reconstruction. For scripted workflows, dependency changes can alter execution. For historical software, obtaining an executable may become the main obstacle. These constraints should be reported as conditions of analytical action rather than treated as incidental administrative details. They determine whether a documented method can actually be repeated.

6. PROPOSED EMPIRICAL EVALUATION

The framework becomes testable through a controlled follow-up study. Such a study should assemble a legally usable corpus with documented sources, genre and language variety. Rather than claim coverage of all Arabic, it could begin with one defined domain, such as contemporary written news. A second domain could then test transfer. Corpus size should follow the evaluation tasks and annotation resources. Reference annotation should use explicit guidelines for token boundaries, lemma selection and uncertain cases. Independent Arabic specialists should annotate a subset, discuss disagreements and retain unresolved alternatives where justified. This establishes a documented basis for evaluating outputs while recognising that linguistic ambiguity cannot always be reduced to one unquestioned label or automatic decision.

The evaluation should separate shared representation from native processing. In the first condition, tools receive equivalent prepared data, allowing comparison of retrieval, presentation and statistical outputs. In the second, each tool uses its available native or configured processing, allowing evaluation of the complete workflow. Morphological processors would be paired with a declared corpus environment rather than penalised for lacking a graphical concordancer. Unsupported tasks would be marked inapplicable. This prevents a comparison from confusing different software purposes with differences in quality. Retrieval evaluation would compare returned occurrences with the reference set and calculate precision, recall and appropriate combined measures. Morphological evaluation would examine segmentation boundaries and selected annotations separately. Output comparisons would investigate rank changes and contextual evidence rather than rely solely on total counts. Usability tasks could ask researchers to retrieve a specified construction, explain a frequency discrepancy and reconstruct a prior analysis. Success, time and error categories would then provide evidence of actualised affordances rather than undocumented impressions of ease.

Sensitivity analysis would vary normalization, diacritic policy, segmentation and comparison settings while holding the source corpus constant. The objective would be to determine which substantive interpretations survive reasonable analytical alternatives. If rankings change but contextual conclusions remain stable, the study should explain why. If an interpretation changes with a minor preprocessing decision, that dependence becomes a result worth reporting. Software evaluation would thereby move beyond selecting a winner and identify how particular decisions reshape evidence for particular Arabic research questions.

7. DISCUSSION

The review's central contribution is the shift from compatibility as a product label to compatibility as a task dependent relationship. Affordance theory explains this shift because documentary feature lists cannot explain actual analytical success without considering researchers and corpora. It also requires validation because the existence of a possible action does not establish that its output is accurate. The framework therefore connects theory and method through observable operations, providing a basis for evaluating what researchers can accomplish and how securely their conclusions follow. For Arabic language education, the same distinction protects pedagogical claims. Concordances can provide materials for examining vocabulary and grammatical patterns, while segmented representations can make attached elements visible. These are potential

License: This is an open access article under the CC BY 4.0 license: https://creativecommons.org/licenses/by/4.0/

instructional uses. Demonstrating improved understanding or speaking performance would require learner data and an appropriate assessment design. Software documentation cannot establish such outcomes. The theoretical connection to van Lier (2004) therefore concerns the conditions under which learners or teachers might perceive and use language resources, with educational effectiveness reserved for subsequent empirical investigation.

The main limitation is the selective secondary evidence base. No common corpus was processed across the reviewed systems, and no user study was conducted. The review cannot establish comparative accuracy, speed or usability. Selection is purposive, so omitted tools may provide relevant alternatives. Historical and current evidence also have different strengths. These limits do not remove the methodological contribution, but they define it: the article offers a critical synthesis and an evaluation framework whose empirical propositions remain open to testing.

8. CONCLUSION

Arabic compatible word analysers support different stages of corpus research, and their suitability depends on the relationship among technical functions, linguistic representation, researcher competence and analytical purpose. AntConc, WordSmith, LancsBox and Sketch Engine primarily support corpus exploration, whereas Buckwalter, SAMA, MADAMIRA and CAMeL Tools contribute morphological processing in different ways. aConCorde illuminates Arabic centred concordancer design, while Sakhr remains a historical comparator with unresolved current specifications. Affordance theory explains these capabilities relationally; validation determines whether their use supports defensible findings. The proposed evaluation framework provides a researchable pathway from documented functionality to tested analytical action, replacing an unsupported universal ranking with transparent, task specific evidence.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest

FUNDING STATEMENT

No funding was received for this research

REFERENCES

1) Alfaifi, A., & Atwell, E. (2016). Comparative evaluation of tools for Arabic corpora search and analysis. *International Journal of Speech Technology, 19*(2), 347–357. https://doi.org/10.1007/s10772-015-9285-5
2) Al-Sughaiyer, I. A., & Al-Kharashi, I. A. (2004). Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology, 55*(3), 189–213. https://doi.org/10.1002/asi.10368
3) Anthony, L. (2026). *AntConc (Version 4.4.2)* [Computer software]. Waseda University. https://www.laurenceanthony.net/software/antconc/
4) Brezina, V. (2018). *Statistics in corpus linguistics: A practical guide*. Cambridge University Press. https://www.cambridge.org/core/books/statistics-in-corpus-linguistics/4E530F86B328B2287681AD240796D2CF
5) Brezina, V., McEnery, T., & Wattam, S. (2015). Collocations in context: A new perspective on collocation networks. *International Journal of Corpus Linguistics, 20*(2), 139–173. https://doi.org/10.1075/ijcl.20.2.01bre
6) Brezina, V., & Platt, W. (2026). *LancsBox X* [Computer software]. Lancaster University. https://lancsbox.lancs.ac.uk/
7) Buckwalter, T. (2004). *Buckwalter Arabic Morphological Analyzer version 2.0 (LDC2004L02)* [Software and lexical resources]. Linguistic Data Consortium. https://doi.org/10.35111/050q-5r95
8) CAMeL Lab. (n.d.-a). *Camel_tools.tokenizers.morphological* [Software documentation]. Retrieved October 2, 2026, from https://camel-tools.readthedocs.io/en/latest/api/tokenizers/morphological.html
9) CAMeL Lab. (n.d.-b). *camel_tools.utils.normalize* [Software documentation]. Retrieved September 29, 2026, from https://camel-tools.readthedocs.io/en/latest/api/utils/normalize.html
10) Grant, M. J., & Booth, A. (2009). A typology of reviews: An analysis of 14 review types and associated methodologies. *Health Information & Libraries Journal, 26*(2), 91–108. https://doi.org/10.1111/j.1471-1842.2009.00848.x
11) Habash, N. Y. (2010). *Introduction to Arabic natural language processing*. Morgan & Claypool. https://doi.org/10.1007/978-3-031-02139-8
12) Hutchby, I. (2001). Technologies, texts and affordances. *Sociology, 35*(2), 441–456. https://doi.org/10.1177/S0038038501000219
13) Lexical Computing. (n.d.). *Arabic text corpora*. Sketch Engine. Retrieved October 1, 2026, from https://www.sketchengine.eu/corpora-and-languages/arabic-text-corpora/
14) Maamouri, M., Graff, D., Bouziri, B., Krouna, S., Bies, A., & Kulick, S. (2010). *LDC Standard Arabic Morphological Analyzer version 3.1(LDC2010L01)* [Software and lexical resources]. Linguistic Data Consortium. https://doi.org/10.35111/wgjk-zy44
15) Obeid, O., Zalmout, N., Khalifa, S., Taji, D., Oudah, M., Alhafni, B., Inoue, G., Eryani, F., Erdmann, A., & Habash, N. (2020). CAMeL Tools: An open source Python toolkit for Arabic natural language processing. *Proceedings of the Twelfth Language Resources and Evaluation Conferenc*e (pp. 7022–7032). European Language Resources Association. https://aclanthology.org/2020.lrec-1.868/
16) Pasha, A., Al-Badrashiny, M., Diab, M., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., & Roth, R. (2014). MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. *Proceedings of the Ninth International Conference on Language Resources and Evaluation* (pp. 1094–1101). European Language Resources Association. https://aclanthology.org/L14-1479/
17) Roberts, A., Al-Sulaiti, L., & Atwell, E. (2006). aConCorde: Towards an open-source, extendable concordancer for Arabic. *Corpora, 1*(1), 39–60. https://doi.org/10.3366/cor.2006.1.1.39

License: This is an open access article under the CC BY 4.0 license: https://creativecommons.org/licenses/by/4.0/

Corresponding Author: Zainur Rijal Abdul Razak

18) Scott, M. (n.d.). WordSmith Tools version 9 documentation [Software documentation]. Retrieved October 11, 2026, from https://lexically.net/downloads/version9/HTML/overview.html
19) Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research, 104*, 333–339. https://doi.org/10.1016/j.jbusres.2019.07.039
20) Unicode Consortium. (2024). Middle East I. In The Unicode Standard, version 16.0 (Chapter 9). https://www.unicode.org/versions/Unicode16.0.0/core-spec/chapter-9/
21) van Lier, L. (2004). *The ecology and semiotics of language learning: A sociocultural perspective*. Kluwer Academic Publishers. https://doi.org/10.1007/1-4020-7912-5

License: This is an open access article under the CC BY 4.0 license: https://creativecommons.org/licenses/by/4.0/

Corresponding Author: Zainur Rijal Abdul Razak