The Journal of Open Source Software

DocuScope Corpus Analysis & Concordancer: A Streamlit Application for Rhetorical and Linguistic Text Analysis

David West Brown [1¶]

1 Carnegie Mellon University, Department of English ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 19 October 2025
Published: unpublished

Summary

DocuScope Corpus Analysis & Concordancer is a Streamlit application for corpus and rhetorical text analysis. It combines spaCy linguistic annotation with DocuScope rhetorical tagging—a taxonomy that identifies functional language patterns such as narrative, reasoning, and description—and runs in either desktop or multi-user modes. A headless API and CLI allow scripted workflows without the web interface.

Version 0.4.1 of the software is archived on Zenodo (doi:10.5281/zenodo.17392153) (Brown, 2025).

Statement of need

Corpus linguistics and computational text analysis are established methods in linguistics, writing studies, and digital humanities (Biber, 2011; McEnery & Hardie, 2012). However, existing tools present researchers with a fragmented landscape that forces compromises between accessibility and analytical depth.

The DocuScope rhetorical taxonomy (D. S. Kaufer et al., 2004) addresses systematic rhetorical analysis, identifying functional language patterns beyond surface-level linguistic features. However, DocuScope's established implementations relied on rule-based string matching, limiting integration with modern NLP pipelines and restricting adoption outside specialized research groups with access to proprietary tools. This barrier is particularly problematic in educational contexts, where students and novice researchers need access to authentic corpus analysis without first mastering programming or command-line interfaces.

DocuScope CA addresses this gap by unifying DocuScope's hierarchical rhetorical tagging with contemporary linguistic annotation, transparent provenance tracking, flexible deployment options, and educational accessibility in a single open-source package. The intuitive web interface enables students and novices to conduct sophisticated corpus analysis without programming prerequisites, while the API/CLI supports reproducible research workflows for advanced users.

State of the field

Established tools like AntConc (Anthony, 2005) excel at concordancing, frequency analysis, and keyword identification but provide no part-of-speech or rhetorical annotation capabilities. Web-based platforms like Voyant Tools (Sinclair & Rockwell, 2016) offer accessible text visualization and basic analysis but similarly lack linguistic tagging and rhetorical analysis features. Code-centric frameworks (spaCy, NLTK) provide sophisticated linguistic processing but require substantial programming expertise and offer no built-in rhetorical analysis. Proprietary tools often combine features but lack transparency, reproducibility controls, and flexible deployment options.

Creating new software was necessary because existing corpus tools lacked the architectural capacity for integrating rhetorical tagging with modern NLP pipelines while maintaining educational accessibility. AntConc lacks extensibility for custom trained models; Voyant Tools runs exclusively in browser contexts incompatible with multi-stage NLP pipelines; spaCy and NLTK require programming expertise that would exclude our educational audience. Contributing DocuScope functionality to any single existing tool would either sacrifice the dual mandate of research-grade performance and educational accessibility, or require fundamental architectural changes incompatible with those projects' design goals. The solution required separating processing logic from presentation to enable the same analytical core to serve interactive learners (Streamlit UI), reproducible research scripts (API/CLI), and diverse deployment contexts (desktop, web, container, hosted)—a design philosophy not aligned with existing tools' architectures.

Software Design

DocuScope CA builds upon two decades of DocuScope rhetorical taxonomy development (D. S. Kaufer et al., 2004; D. Kaufer & Ishizaki, 2023) by modernizing the framework from rule-based string matching to trained spaCy models, dramatically expanding reach and accessibility. This represents the first open-source implementation integrating DocuScope rhetorical tagging with spaCy's linguistic pipeline through custom trained models. The work extends rather than replaces the existing DocuScope ecosystem: the rhetorical dictionaries and linguistic theory remain foundational, while the technical implementation enables integration with contemporary NLP infrastructure and open-source distribution.

The architecture separates processing logic from presentation, enabling the same analytical core to serve interactive learners (Streamlit UI), reproducible research scripts (API/CLI), and diverse deployment contexts (desktop, web, container, hosted). This separation matters because it allows researchers to move fluidly between exploratory interface-driven discovery and reproducible programmatic workflows without switching tools or losing analytical continuity. An explicit provenance manifest captures software version, model identifiers, content hashes, and processing parameters, ensuring reproducible analysis across different deployment modes.

Key trade-offs included choosing Polars over Pandas (prioritizing performance for large corpora over ecosystem maturity), Streamlit over Flask/Django (rapid development and lower maintenance burden over fine-grained UI control), and bundling pre-trained models (immediate accessibility for reviewers and students over minimal package size). These decisions reflect the dual mandate of research-grade performance and educational accessibility, where ease of adoption matters as much as analytical capability.

Implementation

The software is built on Python 3.11 with spaCy (Honnibal et al., 2020) for linguistic processing, Polars (Vink, 2023) for high-performance columnar data operations, Streamlit (Snowflake Inc., 2023) for the web interface, and Plotly for interactive visualizations. The `docuscospacy` package integrates DocuScope rhetorical tagging into the spaCy pipeline. All core functionality operates offline with bundled models; external API keys are required only for optional AI-assisted analysis features. Comprehensive tests exercise parsing accuracy, session persistence, and analysis workflows.

Ecosystem

DocuScope CA operates within a broader ecosystem designed for textual analysis. The architecture centers on the `docuscospacy` Python package, which extends spaCy with DocuScope rhetorical tagging capabilities. Pre-trained models are distributed via HuggingFace Hub, built from curated training datasets also available on HuggingFace, ensuring transparent model provenance and reproducibility.

This ecosystem supports multiple deployment modes: the web application (this paper), a cross-platform desktop application, and headless API/CLI access. The web application prioritizes educational accessibility and collaborative research, while the desktop version serves individual researchers requiring offline capabilities.

The layered design separates processing logic from interface concerns. Core functions handle corpus ingestion, spaCy+DocuScope parsing, and metric computation, with results cached by content hash to avoid redundant processing.

Usage and reproducibility

Users may deploy via hosted instance, local container, desktop application, or headless API/CLI. A sample corpus and reproducible script (`paper/scripts/run_example.py`) generate deterministic outputs including token annotations, frequency tables, tag distributions, and a provenance manifest capturing software version, model identifiers, content hashes, and corpus statistics. These artifacts can be regenerated to validate analytical results.

Interactive workflow

The typical interactive workflow demonstrates how students and researchers can conduct sophisticated corpus analysis without programming knowledge: (a) select from built-in sample corpora or upload custom text collections; (b) process the corpus through the integrated spaCy+DocuScope pipeline to generate token-level linguistic and rhetorical annotations; (c) process metadata (encoded into file names); (d) explore frequency distributions across tokens, part-of-speech tags, and rhetorical categories; (e) apply filters, create visualizations, and export results for statistical analysis. This workflow supports exploratory discovery and hypothesis-driven research while maintaining provenance tracking.

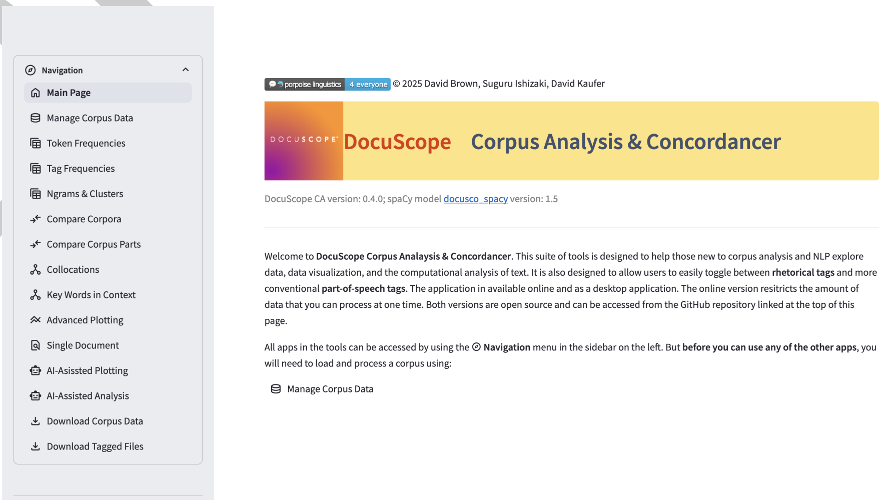

Figure 1: Landing page showing primary navigation menu and real-time processing status indicators for corpus analysis workflows.

The Journal of Open Source Software

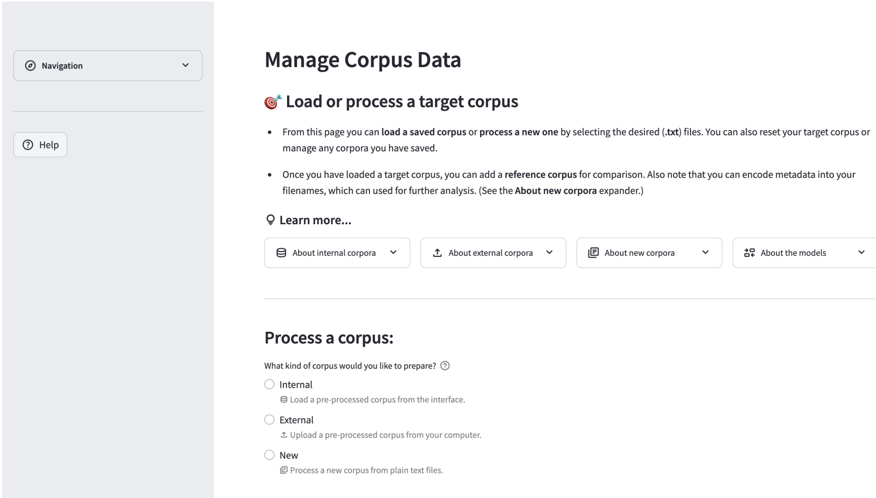

Figure 2: Corpus management interface allowing users to select from internal sample datasets or upload custom text collections with automatic format detection.

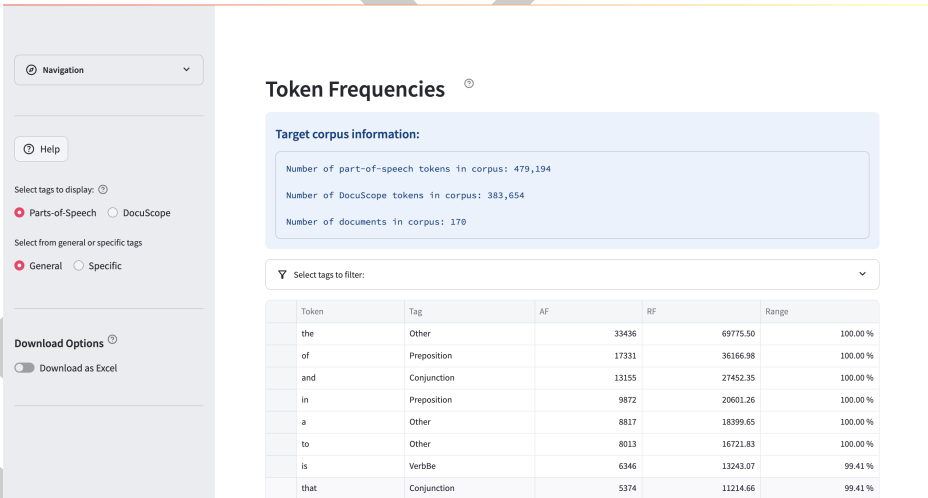

Figure 3: Token frequency analysis displaying sortable, filterable tables of word frequencies with part-of-speech and rhetorical tag annotations, ready for download.

The Journal of Open Source Software

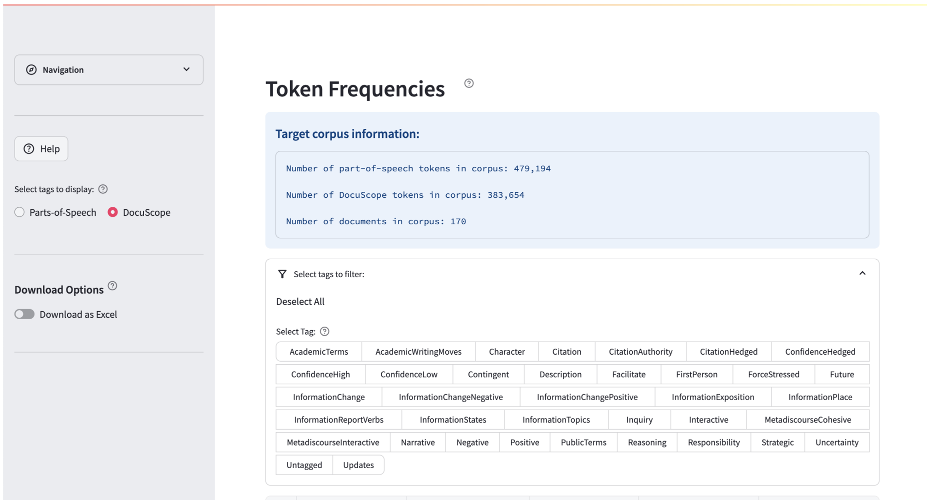

Figure 4: Advanced filtering interface enabling users to refine analysis by applying multiple criteria to focus on specific linguistic or rhetorical patterns of interest.

Performance

Benchmark (50 docs; 132k words; Python 3.11.8; 8-core, 24 GB RAM) achieved ~5.6 documents/s (~890k words/min steady state, 1.1 min per million words) excluding initial model load. Contributing factors: batched spaCy calls, vectorized Polars group-bys, minimal intermediate serialization, and hash-based avoidance of duplicate work.

Research Impact Statement

DocuScope CA demonstrates realized impact through institutional deployment and ecosystem integration. The software serves approximately 500 first-year students per semester in a Carnegie Mellon writing course focused on data-driven analysis, with additional usage via a hosted enterprise instance (docuscope-ca.eberly.cmu.edu). Cross-platform desktop applications enable offline usage. The underlying `docuscospacy` package receives 150-200 monthly PyPI downloads, indicating broader adoption. Pre-trained models distributed via HuggingFace Hub (browndw/en_docusco_spacy) establish transparent provenance and facilitate ecosystem integration.

The software provides novel capability by unifying DocuScope rhetorical analysis—previously available only through proprietary tools—with modern open-source NLP infrastructure in an accessible package. This combination enables corpus-based rhetorical analysis at scale, as demonstrated in published research utilizing the framework (Brown & Laudenbach, 2022; Wetzel et al., 2021) and the recent edited volume on DocuScope applications (Brown & Wetzel, 2023).

Community-readiness is evidenced through comprehensive testing (unit, integration, performance, and UI tests), Apache 2.0 licensing, extensive documentation, multiple deployment modes, reproducible benchmark workflows, and formal citation metadata (CITATION.cff) with Zenodo archival (doi:10.5281/zenodo.17392153). The separation of processing logic from interface enables integration into diverse workflows, from classroom instruction to large-scale corpus studies, addressing methodological gaps in digital humanities and corpus linguistics that existing fragmented tools leave unresolved.

AI Usage Disclosure

DocuScope CA was developed over a three-year period beginning in 2022, prior to the widespread availability of AI-assisted coding tools. The project has maintained public repositories since its inception, beginning with an initial GUI wrapper (DocuConc, https://github.com/browndw/DocuConc) before evolving to the current Streamlit-based architecture. The software architecture, core processing pipeline, and user interface were designed and implemented without generative AI assistance. The software itself includes optional AI-assisted analysis features (utilizing the OpenAI API) that users may enable for exploratory data analysis; these features are clearly documented as experimental and optional. This paper was written without the use of generative AI tools for content generation or authoring.

Acknowledgements

I acknowledge the DocuScope team at Carnegie Mellon University for the rhetorical framework, the spaCy development team for NLP infrastructure, and the Streamlit team for the web framework. This work received no external funding.

References

Anthony, L. (2005). AntConc: Design and development of a freeware corpus analysis toolkit for the technical writing classroom. *IPCC 2005. Proceedings. International Professional Communication Conference, 2005.*, 729–737. https://doi.org/10.1109/IPCC.2005.1494244

Biber, D. (2011). Corpus linguistics and the study of literature: Back to the future? *Scientific Study of Literature*, *1*(1), 15–23. https://doi.org/10.1075/ssol.1.1.02bib

Brown, D. W. (2025). *DocuScope corpus analysis & concordancer (v0.4.1)* (Version 0.4.1). Zenodo. https://doi.org/10.5281/zenodo.17392153

Brown, D. W., & Laudenbach, M. (2022). Stylistic variation in email. *Register Studies*, *4*(1), 1–29. https://doi.org/10.1075/rs.20023.bro

Brown, D. W., & Wetzel, D. Z. (Eds.). (2023). *Corpora and rhetorically informed text analysis: The diverse applications of DocuScope* (Vol. 109). John Benjamins Publishing Company. https://doi.org/10.1075/scl.109

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). *spaCy: Industrial-strength natural language processing in Python* (Version 3.x). Explosion. https://spacy.io

Kaufer, D. S., Ishizaki, S., Butler, B. S., & Collins, J. (2004). *The power of words: Unveiling the speaker and writer's hidden craft*. Routledge. https://doi.org/10.4324/9781410609748

Kaufer, D., & Ishizaki, S. (2023). The DocuScope project: History, theory and future directions. In *Corpora and rhetorically informed text analysis* (pp. 2–24). John Benjamins Publishing Company. https://doi.org/10.1075/scl.109.01kau

McEnery, T., & Hardie, A. (2012). *Corpus linguistics: Method, theory and practice*. Cambridge University Press. https://doi.org/10.1017/CBO9780511981395

Sinclair, S., & Rockwell, G. (2016). *Voyant tools*. https://voyant-tools.org

Snowflake Inc. (2023). *Streamlit*. https://streamlit.io

Vink, R. (2023). *Polars*. https://doi.org/10.5281/zenodo.7697217

Wetzel, D., Brown, D., Werner, N., Ishizaki, S., & Kaufer, D. (2021). Computer-assisted rhetorical analysis: Instructional design and formative assessment using DocuScope. *The*

Journal of Writing Analytics, *5*(1), 292–323. https://doi.org/10.37514/JWA-J.2021.5.1.09