Executable claims for graph-based text analysis: from exploration to auditability

Aris Xanthos & Colin Luginbühl
Department of Language and Information Sciences
University of Lausanne
{aris.xanthos,colin.luginbuhl}@unil.ch

Keywords: computational text analysis; digital humanities; graph-based systems; claims; evidence; provenance; reproducibility; sensitivity

1 Introduction

Across computational science, reproducibility is increasingly treated as a baseline standard, supported by engineering practices that make analyses rerunnable via explicit dependencies and automation (Peng, 2011; Wilson et al., 2014). In the particular case of computational text analysis (CTA), practitioners from various fields converge on the need to make core analytic commitments explicit–including what counts as the unit of analysis, how it is tokenized, what preprocessing is applied, which portion (and version) of the data is in scope, and which computational parameters were used–so that results can be meaningfully validated and reproduced (e.g., Maier et al., 2018; Magnusson et al., 2023; Müller-Hansen et al., 2020).

In digital humanities (DH), recent work on replicability highlights persistent obstacles to checking and rerunning published analyses, in particular missing code, incomplete dependency specifications, and insufficient environment documentation (Illmer, 2025). This is compounded by the fact that in DH practice, CTA is often driven by interactive, exploratory tools that enable fast visual iteration–a legitimate strength, but one that can make it harder to keep core analytic commitments explicit and recoverable. Graph-based analysis, in which the analysis specification is inherently inspectable and rerunnable, offers a promising substrate for addressing these concerns (Rockwell and Bradley, 1998). However, rerunnable specifications alone do not make conclusions checkable: they typically do not explicitly represent which outputs count as evidence for which claims, under which scope and assumptions.[1] Our proposal is therefore to extend graph-based analysis with an epistemic layer in which scope, evidence, and claims are typed objects, and claims are executable: they enforce provenance and compatibility constraints, and compute a transparent support status.

2 Representation and execution

2.1 Scope, evidence, and claims

We separate a small execution core (the *kernel*) from one or more authoring surfaces. The kernel executes an operator graph and records analysis state as three explicit object types: **Scope**, **Evidence**, and **Claims**. A scope identifies the declared dataset and slice. Graph nodes persist typed evidence artifacts (e.g., scalar measurements, tables) in an evidence store; downstream nodes and claims refer to stored artifacts by reference. A claim is executable over evidence. It checks that required inputs are present and compatible under the active claim template, including their data grounding, selection predicates, tokenization and normalization, and source-weighting policies. Then it computes the claimed quantity and produces a claim-result artifact with status SUPPORTED, CONTRADICTED, UNDER-SUPPORTED, or NOT-COMPUTABLE.

Provenance metadata is layered. At the context level, artifacts record scope together with tokenization, normalization, and source-weighting policies (the latter defaults to pooled weighting in the pilot but is always recorded). At the artifact level, evidence stores the minimal definition needed for safe reuse; for a proportion (see section 3) this includes numerator and denominator predicates. This aligns with provenance modeling

[1] The same limitation applies to other executable forms, such as computational notebooks: the relevant information may be documented or encoded ad hoc, but is not ordinarily represented explicitly and systematically as part of the executable analysis.

(W3C Provenance Working Group, 2013). Related work represents claims and evidence in reusable provenance or publication objects (Bechhofer et al., 2010; Clark et al., 2014; Al Manir et al., 2021); our focus is exploratory CTA in DH, where the main design problem is to preserve interactive play while making outputs checkable. Our proposal thus supports a staged path from exploration to stronger outputs: exploratory nodes produce explicit evidence; compatible evidence can be combined into checkable claims; and the authoring surface can propose next-step actions (e.g., instantiate missing evidence or add a sensitivity check, cf. section 4) without committing to a single UI.

2.2 Interactive authoring and incremental execution

Our implementation separates the execution layer (CTA Kernel; Xanthos, 2026a) from the authoring surface (CTA Orange; Luginbühl and Xanthos, 2026), implemented as an add-on to Orange (Demšar et al., 2013). The underlying representation is independent of the authoring surface, as illustrated in section 3. Execution is incremental via dependency tracking and caching/invalidation, preserving interactive speed while keeping an auditable trace (Wilson et al., 2014).

3 Pilot case study: repetition in emoji strings

We illustrate the approach on a compact case study. Users first explore emoji strings extracted from the *What's New, Switzerland?* corpus (Xanthos et al., 2024) interactively (Fig. 1); for the present analysis, a source is defined as a `(chat_id, user_id)` pair. A feature table reports, for each string, its frequency, length (`len`), and diversity (`variety`, number of distinct emoji types). Browsing long strings makes a qualitative pattern salient: `variety` often grows more slowly than `len`, suggesting that long strings tend to be repetition-heavy. This exploratory observation motivates a focused, checkable comparison at the first non-trivial length contrast, between lengths 2 and 3.

We operationalize "pure repetition" as `variety==1`, i.e. the string contains a single distinct emoji type. Two **Proportion** nodes compute `ProportionScalar` evidence artifacts (Fig. 2): within each denominator slice (`len==2` and `len==3`), the numerator counts strings satisfying `variety==1`. Each `ProportionScalar` stores the estimate, masses, and provenance information needed to identify its data grounding, denominator selection, tokenization and normalization, source-weighting policy, and numerator predicate.

With two compatible proportions available, the user can instantiate a comparative-claim template that reuses the stored artifacts, thereby stating precisely what is being asserted; for example,

$$\text{share}(\texttt{len==3}) - \text{share}(\texttt{len==2}) \geq \delta_0,$$

where share(`len==k`) denotes the proportion of length-k strings satisfying `variety==1` and δ_0 is a pre-specified minimum difference threshold (set to 0.3 here). Claim execution checks the operands against the template's compatibility constraints, computes the difference, evaluates the predicate, and stores a claim-result artifact. In this example, claim execution yields $\delta \approx 0.416 > \delta_0$ and status SUPPORTED. The claim-result artifact also records an explicit reason and exposes missing inputs or provenance mismatches when support cannot be established (fields shown in Fig. 2, empty here).

These computational and epistemic relations are represented independently of the authoring surface. For example, the comparative claim above is serialized as follows (abridged):

```
{
  "node_id": "claim_compare",
  "op_id": "Claim",
  "params": {
    "mode": "compare",
    "delta0": 0.3
  },
  "inputs": {
    "scalar_A": {
      "upstream_node": "proportion_len3",
      "upstream_port": "scalar"
    },
    "scalar_B": {
      "upstream_node": "proportion_len2",
      "upstream_port": "scalar"
    }
  }
}
```

The same JSON graph specification can be produced interactively in CTA Orange, constructed programmatically with CTA Kernel (e.g., in a notebook), or written directly; equivalent execution requires consistent operator semantics.

4 Sensitivity check: source weighting policy

The comparative claim depends on the choice of a source-weighting policy. The claim node's UI

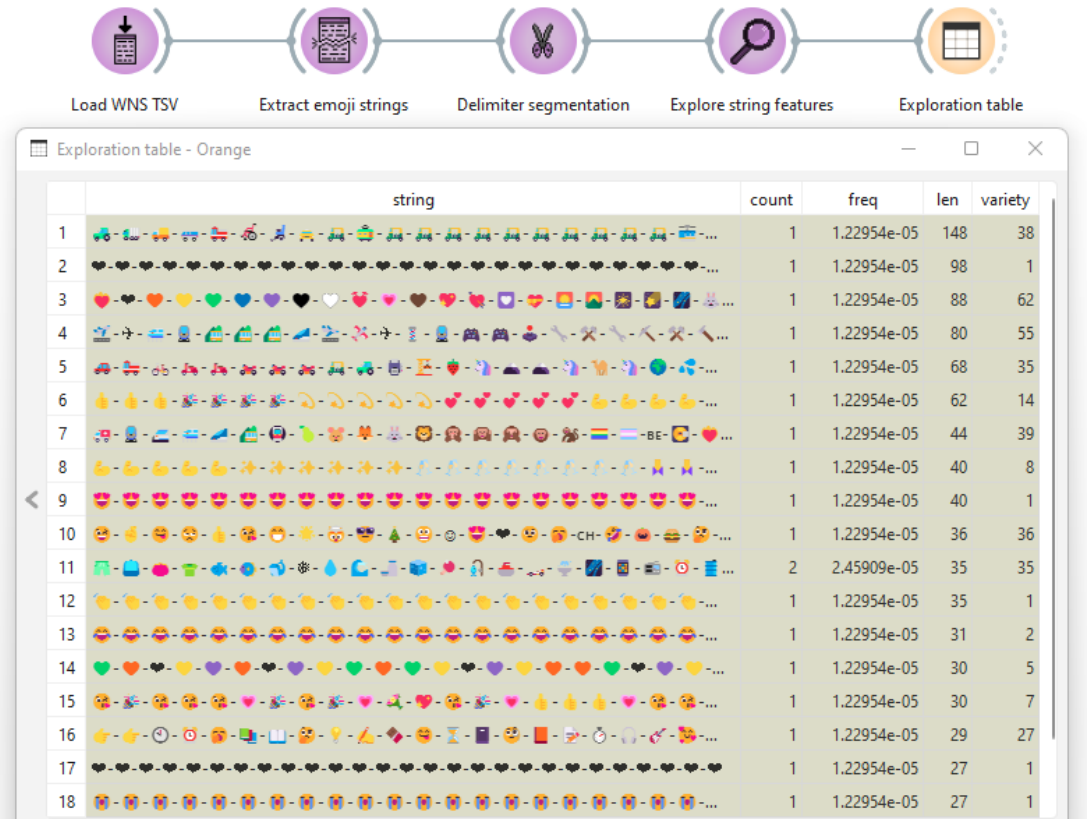

Figure 1: Free exploration in a graph-based UI. The workflow computes string-level features and displays them in a table (`len` = number of emoji tokens; `variety` = number of distinct emoji types). Inspecting long strings shows that `variety` often grows more slowly than `len`, suggesting repetition-heavy strings and motivating a focused comparison.

therefore suggests checking whether the evidence supports the same claim predicate under an alternative source-weighting policy, keeping the data grounding, selection predicates, tokenization and normalization, and δ_0 fixed. In the pilot, the baseline uses a pooled weighting scheme (sources weighted by number of strings); the alternative uses a uniform weighting scheme (each source weighted equally), two estimators discussed in the context of typical-source estimation for imbalanced corpora (Xanthos, 2026b). Under the latter, the estimated difference decreases to $\delta \approx 0.316$, but remains above the pre-specified threshold $\delta_0 = 0.3$. Thus the claim remains supported under this alternative weighting policy, while the attenuation of the estimated difference shows that its magnitude is sensitive to how sources are weighted.

5 Roadmap

The pilot is intentionally small: it demonstrates a portable representation and a concrete path from exploratory views to checkable statements and sensitivity checks. Next steps are to expand the operator and claim-template libraries while keeping primitives compositional, and to support higher-rung claims by adding (i) inferential tests (e.g., uncertainty estimates or resampling-based comparisons) and (ii) annotation workflows that connect evidence to interpretive claims. The software and materials needed to reproduce the paper's evidential core are available in a public replication package (Xanthos, 2026c).

AI assistance statement

ChatGPT 5.1 to 5.6 (OpenAI) and Claude Sonnet 5 (Anthropic) were used as research assistants during the preparation of this paper. Their uses included brainstorming and discussing methodological and software-design alternatives; outlining, drafting, and editing text; and assisting with software specification, implementation, testing, debugging, and documentation. The authors evaluated and decided the substantive design choices and methodological claims, and reviewed, tested, and critically revised

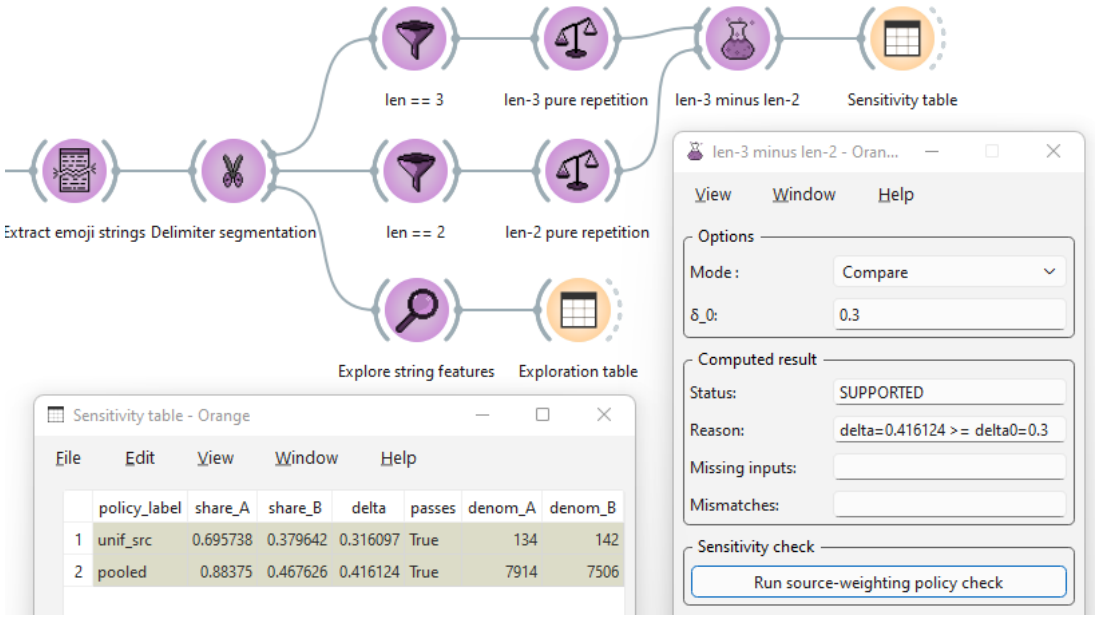

Figure 2: From exploratory evidence to a checkable claim. Two branches filter strings by length (`len==2` vs. `len==3`) and compute the proportion of "pure repetition" (`variety==1`). The claim node evaluates a comparative threshold (δ_0) and records a support status with an explicit reason; the sensitivity check is available as a separate step (results shown in the bottom-left table).

AI-assisted outputs. They remain fully responsible for the content and any errors.

References

Sadnan Al Manir, Justin Niestroy, Maxwell Adam Levinson, and Timothy W. Clark. 2021. Evidence graphs: Supporting transparent and FAIR computation, with defeasible reasoning on data, methods, and results. In Boris Glavic, Vanessa Braganholo, and David Koop, editors, *Provenance and Annotation of Data and Processes*, Springer, Cham, volume 12839 of *Lecture Notes in Computer Science*, pages 39–50. https://doi.org/10.1007/978-3-030-80960-7_3.

Sean Bechhofer, David De Roure, Matthew Gamble, Carole Goble, and Iain Buchan. 2010. Research objects: Towards exchange and reuse of digital knowledge. *Nature Precedings* https://doi.org/10.1038/npre.2010.4626.1.

Tim Clark, Paolo N. Ciccarese, and Carole A. Goble. 2014. Micropublications: a semantic model for claims, evidence, arguments and annotations in biomedical communications. *Journal of Biomedical Semantics* 5:28. https://doi.org/10.1186/2041-1480-5-28.

Janez Demšar, Tomaž Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinovič, Martin Možina, Matija Polajnar, Marko Toplak, Anže Starič, Miha Štajdohar, Lan Umek, Lan Žagar, Jure Žbontar, Marinka Žitnik, and Blaž Zupan. 2013. Orange: Data mining toolbox in Python. *Journal of Machine Learning Research* 14:2349–2353. http://jmlr.org/papers/v14/demsar13a.html.

Viktor J. Illmer. 2025. "Works on my machine": A case study of replicability challenges in computational humanities research. *Anthology of Computers and the Humanities* 3:142–148. https://doi.org/10.63744/iAqEoznkfKuz.

Colin Luginbühl and Aris Xanthos. 2026. CTA Orange (version 0.1.1). https://doi.org/10.5281/zenodo.22235667.

Ian Magnusson, Noah A. Smith, and Jesse Dodge. 2023. Reproducibility in NLP: What have we learned from the checklist? In *Findings of the Association for Computational Linguistics: ACL 2023*. pages 12789–12811. https://doi.org/10.18653/v1/2023.findings-acl.809.

Daniel Maier, Annie Waldherr, Peter Miltner, Gregor Wiedemann, Andreas Niekler, Andreas Keinert, Barbara Pfetsch, Gerhard Heyer, Uli Reber, Thomas Häussler, Hannah Schmid-Petri, and Silke Adam. 2018. Applying LDA topic modeling in communication research: Toward a valid and reliable methodology. *Communication Methods and Measures* 12(2–3):93–118. https://doi.org/10.1080/19312458.2018.1430754.

Finn Müller-Hansen, Max W. Callaghan, and Jan C. Minx. 2020. Text as big data: Develop codes of practice for rigorous computational text analysis in energy social science. *Energy Research & Social Science* 70:101691. https://doi.org/10.1016/j.erss.2020.101691.

Roger D. Peng. 2011. Reproducible research in computational science. *Science* 334(6060):1226–1227. https://doi.org/10.1126/science.1213847.

Geoffrey Rockwell and John Bradley. 1998. Eyecontact: Towards a new design for text-analysis tools. *Digital Studies / Le champ numérique* 1. https://doi.org/10.16995/dscn.232.

W3C Provenance Working Group. 2013. PROV-DM: The PROV data model. W3C Recommendation. W3C Recommendation 30 April 2013. https://www.w3.org/TR/prov-dm/.

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, and Paul Wilson. 2014. Best practices for scientific computing. *PLOS Biology* 12(1):e1001745. https://doi.org/10.1371/journal.pbio.1001745.

Aris Xanthos. 2026a. CTA Kernel (version 0.1.0). https://doi.org/10.5281/zenodo.21872910.

Aris Xanthos. 2026b. Estimating the typical-source distribution in imbalanced corpora. In Antonella Plaia, Mariangela Sciandra, and Alessandro Albano, editors, *18th International Conference on Statistical Analysis of Textual Data*. University of Palermo, Palermo, pages 675–683.

Aris Xanthos. 2026c. Replication package for "executable claims for graph-based text analysis: from exploration to auditability" (version 1.0.0). https://doi.org/10.5281/zenodo.22256624.

Aris Xanthos, Prakhar Gupta, Leyla Benkais, Lliana Doudot, and Andrea Grütter. 2024. What's New, Switzerland? Corpus (version 1.0.0). LaRS – Language Repository of Switzerland. Data set. https://doi.org/10.48656/pa3t-xh52.