Stratum — Cross-Modal Data Intelligence Platform

A design prototype and open invitation to build.

Stratum is a concept for a data analysis platform that takes large text corpora — emails, documents, messages — extracts every keyword from the body text, builds a document-term matrix, discovers co-occurring keyword modules via network analysis, and then uses the timestamps on those modules as a join key into structured numeric datasets (bank transfers, flight logs, purchases, access records) to find cross-modal co-occurrences that neither dataset could reveal alone.

This repository currently contains a **front-end design prototype** (`index.html`) — a fully rendered website describing the platform's intended capabilities. There is no working backend yet. The purpose of making this public is to find collaborators who can validate the analytical approach, identify where the methodology is sound or flawed, and help build toward a real implementation.

What the platform proposes to do

Phase 1 — Text corpus analysis

1. Ingest a text corpus (emails, documents, records) via API or file upload
2. Extract all keywords from body text using NLP preprocessing: tokenization, stopword removal, lemmatization, optional n-gram generation
3. Filter vocabulary by document frequency thresholds to control matrix size
4. Build a **document-term matrix** (rows = documents, columns = extracted keywords, cells = TF-IDF weights)
5. Compute column-wise correlations to produce a **term co-occurrence network**
6. Run community detection (Louvain) to identify **keyword modules** — clusters of terms that co-occur significantly across the corpus without being pre-defined by the analyst
7. Validate discovered modules using Structural Equation Modeling (SEM) — treating each module's keyword set as indicators of a latent variable

Phase 2 — Cross-modal temporal fusion

8. Extract the timestamps of documents belonging to each keyword module, creating a **module activity time series**
9. Accept a second, structured dataset (e.g. bank transfers, flight logs, purchases) that also has timestamps
10. Fuse the two datasets by time using five analytical modes:
    - **Temporal Overlay** — visual co-occurrence of module activity and event frequency

- **Lagged Cross-Correlation** — does text activity lead or follow structured events, and by how long?
- **Event-Driven Windowing** — what structured events cluster inside keyword module activity windows?
- **Shared Entity Linkage** — named entities appearing in both datasets as direct semantic bridges
- **Joint SEM** — both modalities as indicators of the same latent construct

---

What exists right now

Component	Status
Website / design prototype (`index.html`)	✅ Complete
NLP preprocessing pipeline	❌ Not built
Document-term matrix construction	❌ Not built
Term co-occurrence network computation	❌ Not built
Louvain community detection	❌ Not built
SEM implementation	❌ Not built
Module activity time series extraction	❌ Not built
Cross-modal temporal fusion engine	❌ Not built
Visualizations (network graph, timeline, heatmap)	❌ Not built (mocked in HTML)
API connectors (email, database, REST)	❌ Not built

Where we need help

This project needs people who can evaluate the methodology, identify problems with it, and contribute to building it. Specific areas:

Statisticians / Psychometricians

The SEM component is the most methodologically complex and the most likely to be implemented naively. Key open questions:

- Is using network-detected keyword modules as CFA indicator sets for latent variables methodologically sound, or does it introduce circularity?
- What sample size (number of documents) is required for stable SEM estimation at this vocabulary scale?
- Which fit indices are most appropriate given the sparse, non-normal nature of TF-IDF-weighted data?
- Should the SEM be estimated on the raw DTM, a reduced representation (e.g. PCA scores), or the module membership scores?
- Is a WGCNA-style (weighted gene co-expression network analysis) approach more appropriate than Louvain for this data structure?

NLP / Computational Linguists

- What preprocessing choices matter most for downstream network structure (stemming vs. lemmatization, bigram inclusion thresholds, TF-IDF variant)?
- At what vocabulary size does the co-occurrence network become too dense to yield meaningful modules?
- Are there better alternatives to TF-IDF weighting for this use case (BM25, PMI, PPMI)?
- How should named entity recognition be integrated to support the Shared Entity Linkage fusion mode?

Data Engineers / Backend Developers

- The DTM at 10M documents × 100K terms is too large for dense representation — what sparse matrix stack is most appropriate (scipy.sparse, Polars, DuckDB)?
- What network library is best suited for this scale (NetworkX, igraph, graph-tool)?
- Architecture for the cross-modal join: how should the temporal windowing be implemented efficiently when one dataset has millions of timestamped rows?

Domain Experts

If you work in forensic accounting, fraud detection, intelligence analysis, clinical informatics, or computational social science — these are the fields where cross-modal temporal fusion of text and structured records is most likely to have real-world value. We want to know: does the approach make sense in your domain, and what would a meaningful test dataset look like?

Suggested tech stack (open to challenge)

Layer	Proposed	Alternatives welcome
NLP preprocessing	spaCy	NLTK, Stanza

Layer	Proposed	Alternatives welcome
DTM construction	scikit-learn `TfidfVectorizer`	Gensim, custom
Sparse matrix ops	scipy.sparse	Polars, DuckDB
Network analysis	igraph + leidenalg	NetworkX, graph-tool
SEM	semopy (Python)	lavaan (R), pySEM
Dimensionality reduction	UMAP + scikit-learn	Other
Time series / cross-correlation	statsmodels	tsfresh, other
Frontend	Vanilla HTML/CSS/JS	Open to React
Backend API	FastAPI	Flask, other

How to run the prototype

The current prototype is a single static HTML file. No build step, no dependencies.

```bash
git clone https://github.com/YOUR_USERNAME/Stratum-Data.git
cd stratum
open index.html   # or just double-click the file
```

That's it. Everything you see is mocked — the matrix values, network graph, and timeline visualizations are all static. They are illustrations of intended output, not computed results.

A note on example statistics and methodological novelty

The numbers appearing throughout the prototype website and design notes — 4,821 emails, 18,340 terms, Module A = 31% of corpus, Module B = 44%, Module C = 18%, CFI = 0.94, RMSEA = 0.048, r = 0.74, lag = +4 hours, "38% of cases", and "<2 seconds for module detection" — are **entirely fictional**. They were invented as illustrative examples during the design process and are not drawn from any real dataset or benchmark. They should not be cited or treated as expected performance figures.

The **cross-modal temporal fusion framework** (the five fusion modes described in Phase 2) is a novel conceptual proposal. It is constructed from well-established statistical building blocks (cross-correlation, time windowing, SEM) but the specific pipeline combining keyword module timestamps with structured numeric

records has not been validated in the literature and does not have a citable source. It is a hypothesis about what might be useful, not a proven method.

Verified sources for the methods referenced

The following are real, citable papers for the established techniques this project proposes to use. These were verified by searching published literature — they are not AI-generated references.

TF-IDF (term frequency–inverse document frequency) Spärck Jones, K. (1972). "A statistical interpretation of term specificity and its application in retrieval." *Journal of Documentation*, 28(1), 11–21. No open-access link to the 1972 paper exists, but it is described here: https://programminghistorian.org/en/lessons/analyzing-documents-with-tfidf

Louvain community detection algorithm Blondel, V.D., Guillaume, J.L., Lambiotte, R. & Lefebvre, E. (2008). "Fast unfolding of communities in large networks." *Journal of Statistical Mechanics*, P10008. https://arxiv.org/abs/0803.0476

Leiden algorithm (recommended improvement over Louvain) Traag, V.A., Waltman, L. & van Eck, N.J. (2019). "From Louvain to Leiden: guaranteeing well-connected communities." *Scientific Reports*, 9, 5233. https://www.nature.com/articles/s41598-019-41695-z

WGCNA — Weighted Gene Co-expression Network Analysis (flagged as a possible alternative to Louvain) Langfelder, P. & Horvath, S. (2008). "WGCNA: an R package for weighted correlation network analysis." *BMC Bioinformatics*, 9, 559. https://link.springer.com/article/10.1186/1471-2105-9-559 *Caveat: WGCNA was designed for gene expression data. No published paper was found applying it to document-term matrices. Whether it transfers to text corpora is an open question.*

semopy — Python SEM package Igolkina, A.A. & Meshcheryakov, G. (2020). "semopy: A Python Package for Structural Equation Modeling." *Structural Equation Modeling: A Multidisciplinary Journal.* DOI: 10.1080/10705511.2019.1704289 Preprint: https://arxiv.org/abs/1905.09376

SEM fit index thresholds (CFI, RMSEA) Hu, L.T. & Bentler, P.M. (1999). "Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives." *Structural Equation Modeling*, 6(1), 1–55. https://www.tandfonline.com/doi/abs/10.1080/10705519909540118 Open-access copy: http://expsylab.psych.uoa.gr/fileadmin/expsylab.psych.uoa.gr/uploads/papers/Hu_Bentler_1999.pdf

Claims with no verifiable source

- Preference for lemmatisation over stemming for interpretability — commonly held in NLP practice but no specific supporting paper was found for the DTM-then-network-analysis use case
- The five cross-modal fusion modes — an original framework constructed for this project, no prior literature found
- All performance and scale figures in the prototype — invented illustrative examples, not benchmarks

---

Contributing

All contributions are welcome. The most valuable ones right now are **critiques** — if the SEM approach is flawed, if the network analysis methodology has a known problem at this scale, or if there is a better-established technique for any part of this pipeline, please open an Issue and explain it.

To contribute:

1. Fork the repository
2. Create a branch (`git checkout -b feature/your-contribution`)
3. Commit your changes
4. Open a Pull Request with a description of what you've added or changed and why

For methodological discussions, open an **Issue** rather than a PR. Tag it with the relevant label (`methodology`, `nlp`, `sem`, `fusion`, `architecture`).

Intended use cases

The platform is designed for analysts working with datasets where the relationship between text communication and real-world events is the object of study. Example domains:

- **Investigative / forensic** — correlating communication records with financial transactions or travel records
- **Organizational research** — understanding how language in internal communications relates to operational decisions
- **Clinical informatics** — fusing clinical notes keyword modules with coded encounter or prescription data
- **Computational social science** — correlating text corpus activity (news, social media, legislative records) with economic or political event data

---

License

MIT License. See `LICENSE` for details.

Contact

Open an Issue on this repository. All design and methodology questions are in scope.