Digital Scholarship in the Humanities, 2024, **39**, 74–96
https://doi.org/10.1093/llc/fqad071
Advance access publication 20 November 2023
Full Paper

All the world's a (hyper)graph: A data drama

Corinna Coupette [1,*], **Jilles Vreeken**[2], **Bastian Rieck** [3]

[1]Max Planck Institute for Informatics, Saarbrücken, Germany
[2]CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
[3]Institute of AI for Health, Helmholtz Munich, Munich, Germany
*Corresponding author. E-mail: coupette@mpi-inf.mpg.de

Abstract

We introduce Hyperbard, a dataset of diverse relational data representations derived from Shakespeare's plays. Our representations range from simple *graphs* capturing character co-occurrence in single scenes to *hypergraphs* encoding complex communication settings and character contributions as hyperedges with edge-specific node weights. By making multiple intuitive representations readily available for experimentation, we facilitate rigorous *representation robustness checks* in graph learning, graph mining, and network analysis, highlighting the advantages and drawbacks of specific representations. Leveraging the data released in Hyperbard, we demonstrate that many solutions to popular graph mining problems are highly dependent on the representation choice, thus calling current graph curation practices into question. As an homage to our data source, and asserting that science can also be art, we present our points in the form of a play.[1]

DRAMATIS PERSONÆ

Authors. } Persons in the Induction.
Reviewer, a reader. }

Professor. } Part of the Community.
Colleague. }

Creature, a curious mind.
Hyperbard, a faun, sovereign of spirits.
Graph, a gentle spirit.

Scene.—*Sometimes in the Community; and sometimes in the forest.*

INDUCTION.

Scene I.—*Between submission and decision.*

Enter Reviewer *and* Authors.

Rev. What is this? Is this not against the rules?
Auth. The columns? These are only simple tables.
They serve to help us implement blank verse.
We introduce a novel dataset,
With full documentation as Appendix.
Raw data stem from all of Shakespeare's plays [14],
We model them as graphs in many ways,
And demonstrate representations matter.
The data readily accessible [6],
All code is publicly available [7].
What follows, to avoid redundancy,
Conveys our main ideas, as you will see
A tragedy in the Community [5].

ACT I.—DATA.

Scene I.—*The forest, in* Creature's *dream.*

Enter Hyperbard, *with a lute.*

Hyp. What beauty are these woods! In every tree
Lives past enshrined and calling the observant.
The devil? Angels lie in all these details.
Look at the fragile bark, the fractal branching,
The posture, parasites—And see the leaves!
Colors, shapes, textures—all varieties.

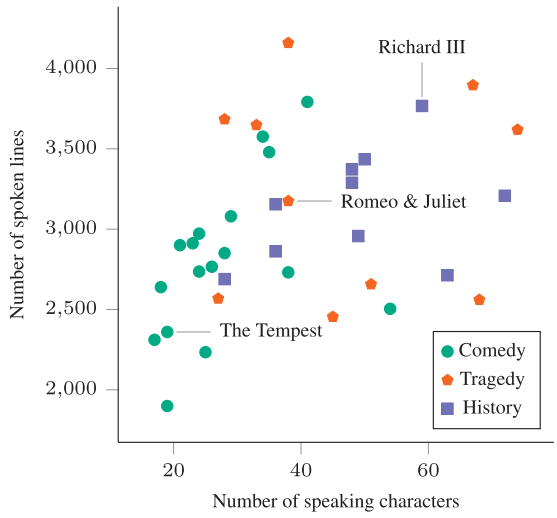

Figure 1. Number of spoken lines versus number of speaking characters in the thirty-seven plays by William Shakespeare. Each point corresponds to a play for which we provide eighteen different (hyper)graph representations.

The fauna—beetles, rodents, insects, birds,
Thriving together in their interaction.
They strike a chord on their lute.
Hyp. From all there is, let there be data!
As data points, we demarcate these trees
And put them into known categories.
They mark the selected trees with leaves of various shapes (Fig. 1).
Hyp. Each tree is full of life, full of relations,
To capture this, we need representations.
They strike another chord. Enter GRAPH.
Gra. You called me, honor?
Hyp. Will you, docile spirit,
Transform these trees to yield discoveries?
Gra. Your honor, master, mistress, sure I can
But there are many different transformations
Among the flurry, which one do you choose?
Hyp. Why choose but one when there exist so many?
How do we even know which one to pick?
Gra. Sir, madam, with respect, your speech is madness!
Did you not call me to produce your truth?
Hyp. What truth? Your transformations are but shadows
Of essence vested with complexity
Cast on the narrow walls of our perception
And varied as you shift and change your light.
Gra. I hear your words but struggle with their meaning.
Which output do you want me to obtain?
Hyp. To every data point associate
A set of transformations as its data.
Such that in all our future inquiries
We treat not only one but many shadows.
Each partly blind, together they create
A truer truth than commonly considered.
Gra. Your honor, as a practicality
We can't enumerate exhaustively.
Among the myriad possibilities
You still will have to choose some transformations.
Hyp. Fair spirit, as an overarching goal,
All our representations should be faithful.
Among the transformations that you see,
How do they differ systematically?
Screaming heard. HYPERBARD *and* GRAPH *vanish*.
CREATURE *wakes*.

SCENE II.—*The Community. In the dining hall.*

COLLEAGUE, *seated at a table. Enter* CREATURE, *carrying a tray*.
Col. Hey fellow, please come join me, have a seat!
CREATURE, *jolted from their thoughts, obeys with reluctance*.
Col. They told me you submitted, so, good cheer!
Awkward silence.
Cre. May I ask you something? Here in the Community, how do you get your data? You hardly go outside. . .
Col. What do you mean? We grab it from the shelves.
There's shelves for almost every data type.
For graphs, e.g., there's OGB [9], and SNAP [12], KONECT [11], and TUD [13], and Netzschleuder [15],
And finally, Network Repository [16].
Cre. Hold on, you are confusing me. How do the graph shelves get their data, then?
Col. You really ask the weirdest things. I guess
They send some hunter-gatherers to catch
Or pick the graphs they find out in the wild.
Cre. You make it sound like graphs exist, for real. But are they not defined by their observers?
Col. Who are you? Not the Spanish Inquisition?
All graphs have nodes and edges, that's what matters.
Sometimes they come with weights or attributes.
Semantics—God, who cares?—graphs are abstractions,
And abstract data is our working truth.
Exeunt.

SCENE III.—CREATURE'*s office.*

In a corner, on the floor, CREATURE, *in contemplation*.
Cre. What canny creatures met my febrile mind. That friendly faun, the gentle spirit, exchanging such profound considerations. I wish I could have stayed a little longer—instead, I'm left to draw my own conclusions. What graph shadows could I create by shining

different lights on what there is? It seems the sensible depends on the semantics.
They close their eyes, following their thoughts.
Cre. When we transform reality to math,
Graphs are but outputs, in—phenomena.
The myriad transformations that we see,
How do they differ systematically?
For now, we shall distinguish three dimensions.
First, our *semantic mapping*—Nodes and edges:
What types of entities do we assign?
Second, our *granularity*—What are
Our modeling units for semantic mapping?
And third, our *expressivity*: What more
Do we attach to all our modeling units?
Directions, weights, and multiplicities,
Or attributes and cardinalities. . .
What universe! *Haec facta, fiant data.*
Tracing coordinate axes with their fingers, they sigh.
Cre. All these distinctions, it appears, are known in the Community [19]. And yet, the knowledge seldom heeded—graph data shelves are filled with all these captive singular truths. We hardly hold what that free faun foresaw: For every data point, a set of transformations as its data. I wonder why.
Exit.

ACT II.—METHODS.

SCENE I.—*The Community.* COLLEAGUE*'s office.*

COLLEAGUE, *trimming a bonsai with scissors.*
Col. Alas, they really want documentation?
CREATURE *steps into the door frame, unnoticed.*
Col. A datasheet [8]? Well—all the world is data,
And all we care for merely data points;
They get created, updated, deleted,
And every data point plays many parts,
Its fate being seven stages. First, *motivation*
Defining purpose or specific tasks.
Then *composition*, sketching the raw data
And telling people where it was obtained,
If anything's amiss. And then *collection*,
How did we get each single data point,
And what else did we check. Then *preprocessing*,
Full of strange quirks and idiosyncrasies,
But made that it looks principled. Then *uses*,
What all things did we do, what could have been,
And what should not be done. Then *distribution*,
If, when, and how will we make data public,
Restrictions by third parties, if imposed,
And also all the laws. Last stage of all,
That ends this template documentary,
Is *maintenance* and hosting and support,
Sans updates, sans errata, sans comment.
CREATURE *retires, flabbergasted.*
COLLEAGUE *stashes the stunted bonsai into a shelf.*
Exit.

SCENE II.—*The forest.*

GRAPH *tending to a mat of moss. On the mat,* CREATURE, *somnolent. Enter* HYPERBARD.
Hyp. So few return once captured by Its magic!
Gra. Playing that dream was worth it, after all.
Cre. Is this a dream no more? Do you exist?
Hyp. Depends on your philosophy. But see,
My GRAPH says you have interesting ideas.
So tell me, how would *you* transform these trees
To bear the fruit of new discoveries?
Cre. Did you not eavesdrop on my ruminations,
Distinguishing between those three dimensions?
Semantic mapping, granularity,
And expressivity—put abstractly?
Hyp. I heard, but what does it all mean in practice?
Cre. Let's walk through an example. Take this tree:
The Tragedy of R. and J.—a play.
When modeled *Les Misérables*-y [10], the nodes
Are characters, and edges—co-occurrence.
That's one semantic mapping, hold this fixed.
Then, as to granularity, we ask
What unit should determine co-occurrence?
The first—most common—option is: a scene.
And here, much modeling ends, unfortunately:
Max simple graphs, min expressivity.
Hyp. But does this not reveal essential structure?
Cre. It smudges all the details, Fig. 2a!
Do the play's namesake heroes co-occur
No more than Montague and Capulet?
Hyp. So should we count-weight edges, Fig. 2b?
Cre. Or introduce edge multiplicity.
The multigraph perspective would allow us
To treat—Fig. 2c—co-occurrence weights.
In our setting, this could, e.g., mean
The count of spoken lines in every scene.
But that is basic expressivity—
We yet have to treat granularity.
To illustrate, in Fig. 3a, we draw
The co-occurrence only for Act III.
The Capulets and Romeo appear
To interact too much—this sparks suspicion.
Hyp. You mean we're introducing information?
Cre. And hiding what there really is to see!
The scene is far too coarse a modeling unit,
Quite often is there movement in between.
We must keep track of entries and of exits
To capture interactions faithfully.
Each part confined by any two such changes,
A *stage group*, separately defines an edge.
Accounting now for expressivity,
These edges may be binary or multi,
Or weighted by lines spoken, Fig. 3b.

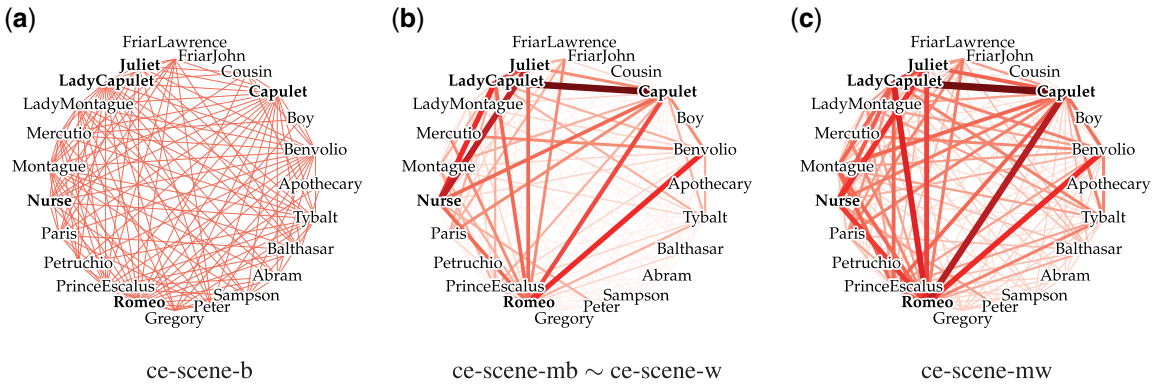

Figure 2. Relationships between the named characters in *Romeo and Juliet* when modeled as binary (ce-scene-b) (**a**), count-weighted (ce-scene-mb ∼ ce-scene-w) (**b**), and line-weighted (ce-scene-mw) (**c**) co-occurrence networks, resolved at the scene level, where we highlight the protagonists appearing in Act III, Scene V. The binary representation is a classic hairball, while the count-weighted representation and the line-weighted representation provide more nuance. In (**c**), the strikingly strong connection between Romeo and Capulet is partly due to Act III, Scene V, where both characters appear but *do not meet* on stage.

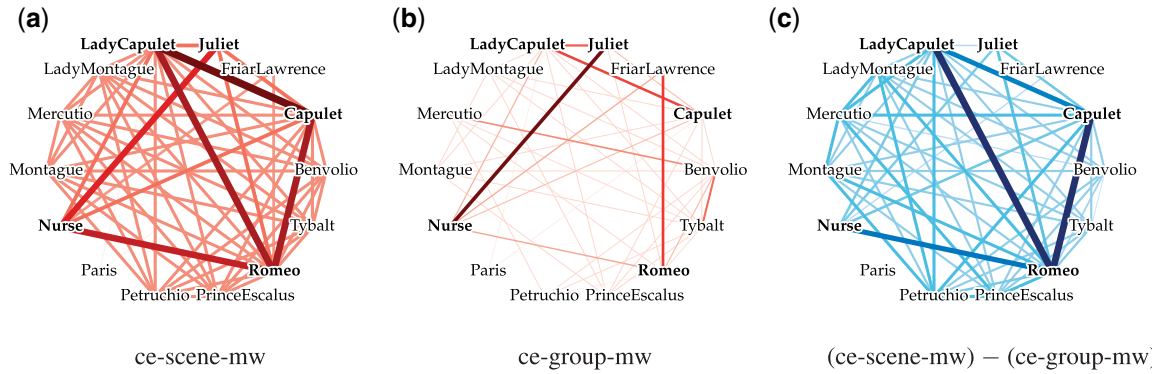

Figure 3. Line-weighted co-occurrence network of the named characters in Act III of *Romeo and Juliet*, resolved at the scene level (ce-scene-mw) (**a**) and at the stage group level (ce-group-mw) (**b**), as well as the difference network between the two [(ce-scene-mw) – (ce-group-mw)] (**c**), where we highlight the protagonists appearing in Act III, Scene V. The coarse-grained representation overestimates the co-occurrence between Juliet's parents (Capulet and Lady Capulet) and Romeo (**a** and **c**), while the fine-grained representation emphasizes Juliet's bond with the Nurse and Romeo's interaction with Friar Lawrence (**b**).

The outcome, evident from Fig. 3c,
Is far from what we had initially.
Thus, even for just one semantic mapping,
And R. and J. as a specific case:
We see at least six decent transformations,
Statistics differing tremendously.
Hyp. So is this all?
Cre. Oh, that is but the start!
Thus far, we've had just characters as nodes.
One possible complaint with this approach
Is that it gives us artificial cliques.
Instead, we could in our semantic mapping
Consider also parts of plays as nodes,
Transforming plays into bipartite graphs,
Whose edges signal character occurrence.
Then granularity, Fig. 4a–b,
Concerns the nodes, but sometimes also edges.
In terms of expressivity, we could
Again attend to weights, and represent
Directionality, see Fig. 4c,
With greater ease than in the one-mode case—
To model single *speech acts*, too, as edges.
Hyp. Now, that is quite a lot—so are you finished?
Cre. Respectfully, the best is yet to come!
Conceptually, all I have just described
Can be derived from a more general model.
All graphs, regarding expressivity
Force '$\in \{1, 2\}$' on cardinality
Of edges—
Hyp. Marvelous mathematically!
Cre. But artificial, thinking critically.
The interactions in your vivid woods—

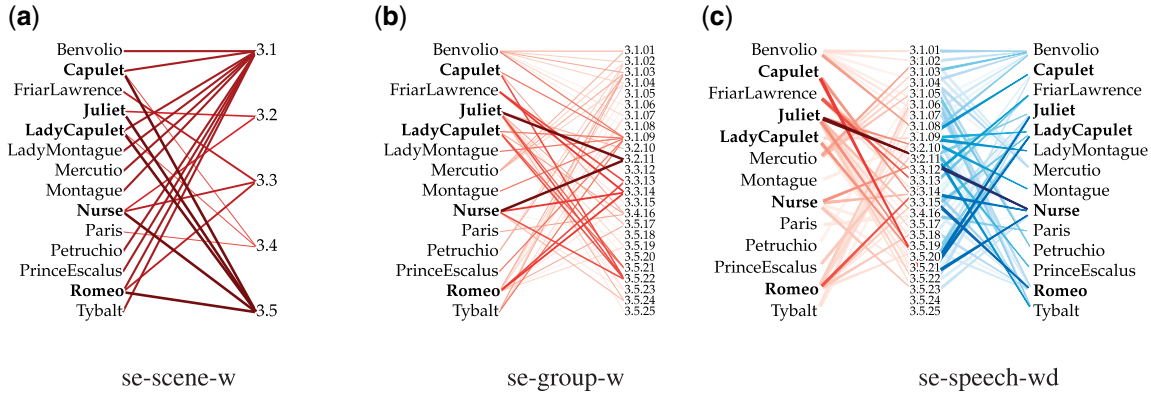

Figure 4. Weighted bipartite graph of named character occurrences in Act III of *Romeo and Juliet*, resolved at the scene level (se-scene-w) (**a**) and at the stage group level (se-group-w) (**b**), as well as the directed weighted bipartite graph resolved at the speech act level, with character nodes split up into speakers and listeners for visual clarity (se-speech-wd) (**c**), where we highlight the protagonists appearing in Act III, Scene V. While the coarse-grained representation overestimates Romeo's role in Act III, Scene V (**a**), the finer-grained representation again highlights Juliet's bond with the Nurse (**b**), and the directed representation reveals the hierarchical structure of their communication (**c**).

Figure 5. Relationship between hypergraphs, clique expansions, and star expansions, illustrated for a toy drama. In the toy drama, characters are capital letters, $\rightarrow X$ denotes entry, $X \rightarrow$ denotes exit, $*$ denotes speech, | marks scene boundaries, ; marks activity boundaries, and, indicates several characters acting together. (**a**) Toy drama, (**b**) Hypergraph ($n = 5, m = 7$), (**c**) Clique expansion ($n = 5, m = 10$), and (**d**) Star expansion ($n = 12, m = 16$).

How many of them are bilateral?
This common cardinality constraint:
Let's do away with it!

Hyp. Then what remains?

Cre. A set system—a *hypergraph*, they say [4],
We visualize its power in Fig. 6.
Confusingly: All graphs are hypergraphs
But not vice versa.

Hyp. Do we need this, GRAPH?

Gra. Well, some found hypergraphs to be quite handy
To capture higher-order interactions [1, 2, 3].
They certainly are more intuitive
Than making cliques of higher arities,
Or else treating relations, too, as nodes.

Cre. We can go far with *graphs* but don't know yet
Just how much further we can get with *hyper*.
Observe the beauty in these hypergraphs:
They readily entail *all* transformations!
From their perspective, what first we discussed
Are *clique expansions*, and our next ideas
Are known as *star expansions* [18]—see, in sum,
Fig. 5, and our proposals in Tab. 1.

Hyp. Things hyper, in their generality,
They seem to suit my woods quite naturally.

Gra. But sovereign, as a practicality,
There's hardly any software letting us
Compute with hypergraphs conveniently!

Hyp. and Cre. [in sync.] Who are you, the Community?

Gra. I'm sorry.

Exeunt.

SCENE III.—*The Community.* CREATURE'*s Office.*

HYPERBARD, *engaging the office plant.*

Gra. [Within] Watch out, they'll be here any minute now!

Enter COLLEAGUE.

Col. Congrats on that acceptance—wait! Who's this?

Hyp. What's in a name? I heard you work with data,
We're colleagues, in a sense—I do the same
But mostly in the wild.

Col. So you're a hunter?

Hyp. Far off! I roam reality's realms
In search of structure that persists across
Perspectives.

Table 1. Overview of relational data representations provided with HYPERBARD for each play attributed to William Shakespeare, based on the TEI simple-encoded XMLs provided by Folger Digital Texts [14]. Unidirectional arrows indicate assignment; bidirectional arrows indicate bijection. We highlight the transformations most commonly used in the literature.

Representation	Semantic Mapping		Granularity	Expressivity
ce-scene-b	Nodes ← Characters Edges ← Co-occurrence		Edges ↔ Scenes	—
ce-scene-mb				Edge order
ce-scene-mw				Edge order, edge weights
ce-group-b			Edges ↔ Stage groups	—
ce-group-mb				Edge order
ce-group-mw				Edge order, edge weights
se-scene-b	Edges ← Occurrence	Nodes (1) ← Characters Nodes (2) ← Play parts	Nodes (2) ↔ Scenes	Partial node and edge order
se-scene-w				Partial node and edge order; edge weights
se-group-b			Nodes (2) ↔ Stage groups	Partial node and edge order
se-group-w				Partial node and edge order; edge weights
se-speech-wd	Edges ← Information flow		Nodes (2) ↔ Stage groups Edges ↔ Speech acts	Partial node order; edge weights, edge directions
se-speech-mwd				Partial node and edge order; edge weights, edge directions
hg-scene-mb	Edges ← Co-occurrence	Nodes ← Characters	Edges ↔ Scenes	Edge order
hg-scene-mw				Edge order, edge weights; edge-specific node weights
hg-group-mb			Edges ↔ Stage groups	Edge order
hg-group-mw				Edge order, edge weights; edge-specific node weights
hg-speech-wd	Edges ← Information flow		Edges ↔ Speech acts	Edge directions, edge weights
hg-speech-mwd				Edge order, edge directions, edge weights

Representation abbreviations follow the pattern <model>-<aggregation>-<properties>, where model ∈ {ce: clique expansion, se: star expansion, hg: hypergraph}, aggregation ∈ {scene: play scene, group: stage group, speech: speech act}, and properties ⊊ {b: binary edges, d: directed edges, m: multi-edges allowed, w: weighted edges}. Binary multigraph representations of clique expansions (ce-∗-mb) can be transformed into weighted graph representations of clique expansions without multiedges (ce-∗-w) using edge counts as weights, but only the multigraph representations can retain order information on edges.

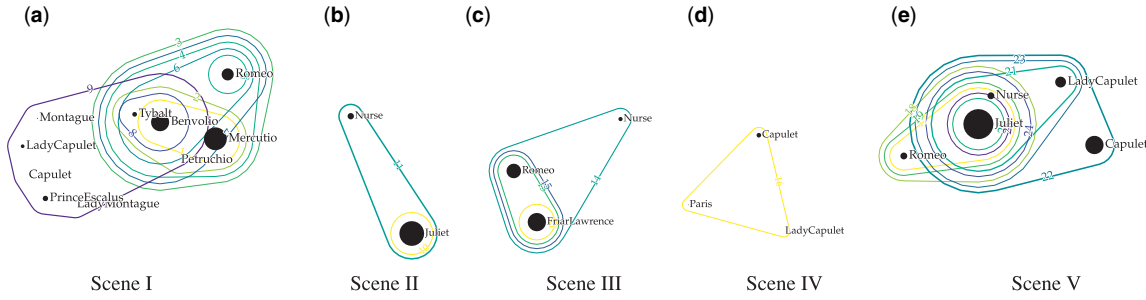

Figure 6. Line-weighted hypergraph resolved at the stage group level, separated by scene and restricted to named characters, for Act III of *Romeo and Juliet*. Edge labels denote stage groups, edge colors indicate edge order, and node sizes and edge widths are proportional to the number of spoken lines. From (**e**), it is visually clear that Romeo never meets Juliet's parents in the scene.

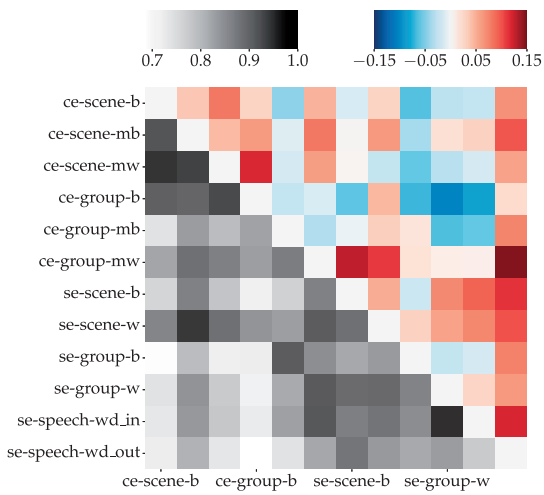

Figure 7. Spearman correlations of degree rankings in the clique and star expansions from Tab. 1 for *Romeo and Juliet* (bottom), and residuals after subtracting the average correlations in the HYPERBARD corpus (top).

Col. By perspectives, you mean tools?
Hyp. I mean representations, as for each
Phenomenon there's many paths to data.
I like to call each path a transformation,
And transformation is my tested trade.
Col. Can you elaborate? What good is that?
Hyp. Let's take a look at, you would say, *graph data*.
Imagine that you have a tree—say, R. and J.—
Col. That famous play?
Hyp. —And that you want to model
The structure of its story as a graph.
Col. Well, obviously, each character's a node
And there's an edge between two nodes in case
They co-occur in more than zero scenes.
Hyp. But this is only one of many options.
And without dwelling on the details here,
Fig. 8 reveals how even simplest things
Such as degree ranks differ with our choices.
The variations vary, too, Fig. 7,
Within a set of trees as data raw.
And—to conclude representation matters—
Less simple transformations may support
More nuanced inquiries as in Fig. 9,
Or exploration over time, Fig. 10.
Col. You worry well, but then, so why should I?
What's in it for my publication record?
Enter PROFESSOR.
Prof. What fool is this?
Col. and Hyp. [in sync.] O that I were a fool!
Enter CREATURE.
Cre. Did you discuss the problem with *the* data?
Hyp. I laid it out for them, to no avail.
Col. You surely got me thinking, but—
Prof. Enough!
My patience is exhausted. Think? Produce!
[To Col.] You, give productive treatment to that thinker.
Exeunt.

ACT III.—OUTLOOK.

SCENE I.—*The Community.* CREATURE*'s Office.*

Enter CREATURE.
A deadline, and a deadline, and a deadline,
Creeps in this petty pace to publication,
To the last syllable of our defense.
They slew my GRAPH and choked my inspiration,
Our work is but a walking shadow thence.
The curiosity that drew me in

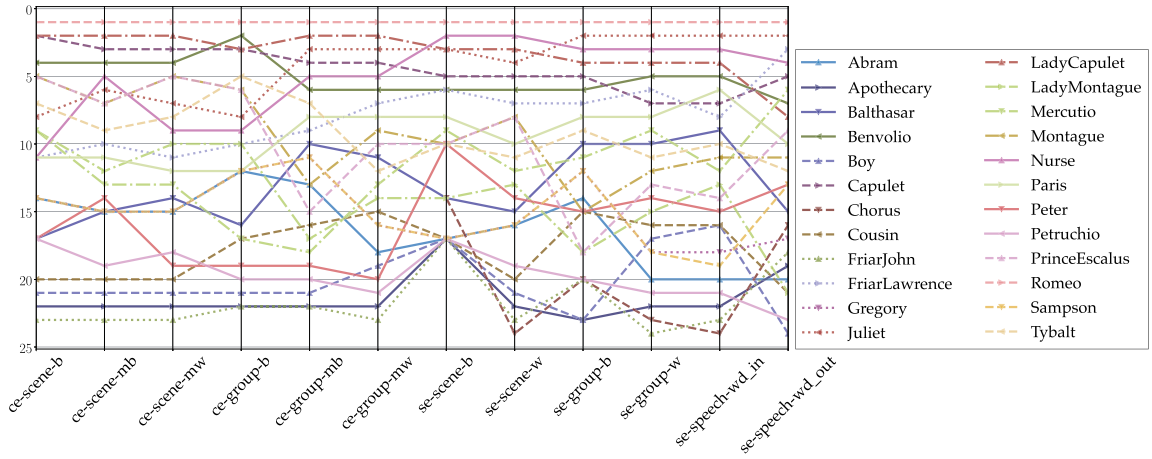

Figure 8. Named characters in *Romeo and Juliet*, ranked by their degree in the clique expansion (ce) and star expansion (se) representations from Tab. 1. We omit the se-speech-mwd representation because its ranking is equivalent to that of the se-speech-wd representation by construction. While Romeo is ranked first under all representations, the rankings differ, inter alia, in the prominence assessment of side characters, such as the Nurse or Friar Lawrence.

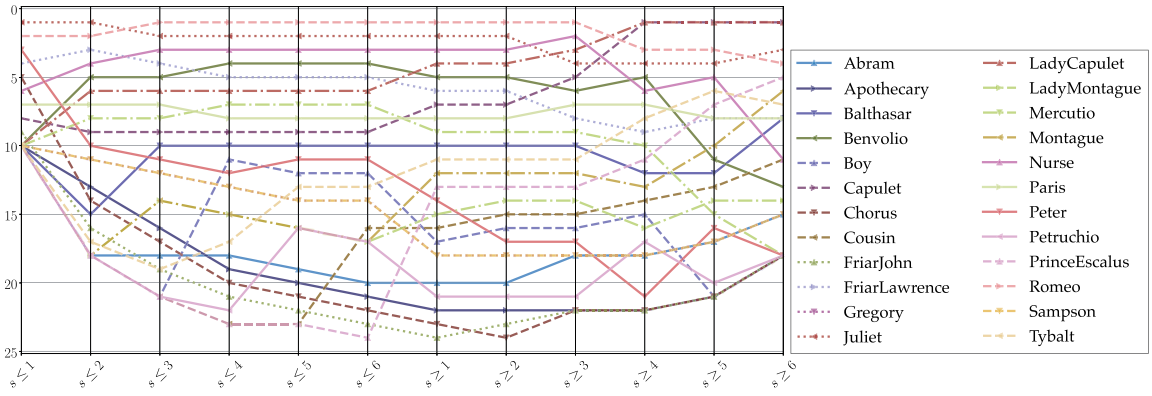

Figure 9. Named characters in *Romeo and Juliet*, ranked by their degree in the weighted hypergraph representation aggregated at the stage group level (hg-group-mw) when considering only hyperedges of cardinality at most s or at least s, for $s \in \{1, 2, 3, 4, 5, 6\}$. Hyperedges of cardinality at most 1 correspond to monologues. While Romeo and Juliet rank highest when including hyperedges of low cardinality, Capulet and Lady Capulet dominate when considering only less private settings.

Now lies in dust. The lofty dreams I had
Of mindful monasterial devotion
To just the cause—no more. Out, out, sore studies!
Should I give up that which I know I love—to save my love for it? And go in silence, not disturbing the Machine? Or should I stay to salvage my beloved—to, once on top, speak out, let nature in?
My story, so it seems, a tragedy
In the Community:
All the world's a (hyper)graph.
Thus, I'll begin.

They write.

1) Graph data does not exist, it is defined.
2) Semantic mapping, granularity, and expressivity are key ingredients to define graph representations.
3) Many phenomena permit several graph representations.
4) Graph data context matters for graph representations.
5) Graph data representations matter for graph methods.
6) Hypergraphs are powerful.
7) HYPERBARD is free.

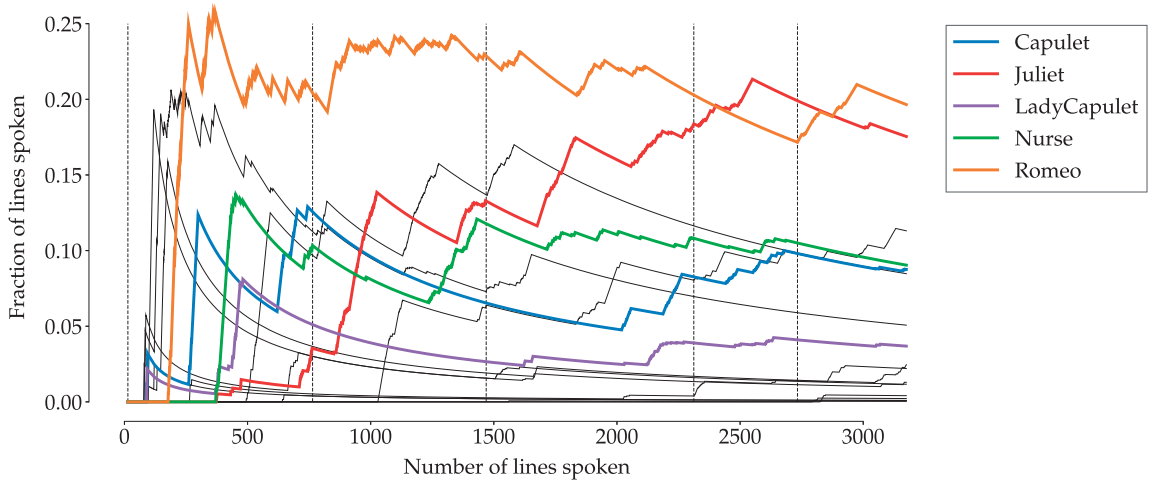

Figure 10. Prominence of named characters in *Romeo and Juliet* over time (excluding named servants), as measured by their fraction of spoken lines, derived from the hypergraph representation resolved at the speech act level (hg-speech-mwd). Dashed vertical lines mark the beginning of each act, and colored lines indicate protagonists of Act III, Scene V. From this perspective, Romeo is most prominent for most of the play, temporarily replaced only by Juliet for a period in Act IV and V.

Author contributions

Corinna Coupette (Conceptualization, Data curation, Investigation, Methodology, Software, Visualization, Writing—original draft, Writing—review and editing), Jilles Vreeken (Supervision, Writing—review and editing), Bastian Rieck (Methodology, Software, Supervision, Validation, Writing—review and editing)

Notes

1. Here, we present only the excerpts of the play that pertain directly to our scientific contributions. For the full story, see the original five-act version available at [5].
2. When construed broadly (as suggested by Gebru et al.), our raw data relates to people because the plays were written by William Shakespeare. The people-specific datasheet questions, however, are ill-suited for our scenario, in which the raw data consists of literary works conceived by someone who died several centuries ago.

References

1. **Aksoy, S. G.** *et al.* (2020) 'Hypernetwork Science via High-Order Hypergraph Walks', *EPJ Data Science*, **9**: 16.
2. **Bai S., Zhang F., and Torr, P. H. S.** (2021) 'Hypergraph Convolution and Hypergraph Attention', *Pattern Recognition*, **110**: 107637.
3. **Battiston, F** *et al.* (2021) 'The Physics of Higher-Order Interactions in Complex Systems', *Nature Physics*, **17**: 1093–98.
4. **Berge, C.** (1989) *Hypergraphs: Combinatorics of Finite Sets*. North-Holland Mathematical Library 45. Amsterdam, The Netherlands: Elsevier.
5. **Coupette, C., Vreeken, J. and Rieck, B.** (2022) *All the World's a (Hyper)Graph: A Data Drama (Extended Version)*. `arXiv: 2206.08225 [cs.LG]`.
6. **Coupette, C., Vreeken, J., and Rieck, B.** (2022) *Hyperbard: (Hyper)Graph Representations of Shakespeare's Plays*. Version 0.0.1. 2022. doi: 10.5281/zenodo.6627159. URL: https://hyperbard.net. [Dataset]
7. **Coupette, C., Vreeken, J., and Rieck, B.** (2022) *Hyperbard: (Hyper)Graph Representations of Shakespeare's Plays*. Version 0.0.1. 2022. doi: 10.5281/zenodo.6627161. URL: https://github.com/hyperbard/hyperbard. [Code]
8. **Gebru, T.** *et al.* (2021) 'Datasheets for Datasets', *Communications of the ACM*, **64**: 86–92.
9. **Hu, W.** *et al.* (2020) 'Open Graph Benchmark: Datasets for Machine Learning on Graphs'. arXiv: `2005.00687`.
10. **Knuth, D. E.** (1994) *The Stanford GraphBase: A Platform for Combinatorial Computing*. New York, NY, USA: ACM Press.
11. **Kunegis, J.** (2013) 'KONECT: The Koblenz Network Collection', in *Proceedings of the International Conference on World Wide Web*, pp. 1343–50.
12. **Leskovec, J. and Sosič, R.** (2016) 'SNAP: A General-Purpose Network Analysis and Graph-Mining Library', *ACM Transactions on Intelligent Systems and Technology (TIST)*, **8**: 1–20.
13. **Morris, C.** *et al.* (2020) 'TUDataset: A Collection of Benchmark Datasets for Learning with Graphs', in *ICML Workshop on Graph Representation Learning and Beyond (GRL+)*. arXiv: 2007.08663. URL: http://www.graphlearning.io.
14. Mowat, B. *et al.*, eds. (2006) *Shakespeare's Plays, Sonnets and Poems*. The Folger Shakespeare Library.

15. **Peixoto, T. P.** (2020) *The Netzschleuder Network Catalogue and Repository*. https://networks.skewed.de/.
16. **Rossi, R. and Ahmed, N.** (2015) 'The Network Data Repository with Interactive Graph Analytics and Visualization', in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 4292–93.
17. **Shakespeare, W.** (1916) *The Complete Works of Shakespeare*. Ed. by Craig W. J.. Oxford, United Kingdom: Oxford University Press.
18. **Srinivasan, B., Zheng, D., and Karypis, G.** (2021). 'Learning over Families of Sets—Hypergraph Representation Learning for Higher Order Tasks', in *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pp. 756–64.
19. **Torres, T.** *et al.* (2021) 'The Why, How, and When of Representations for Complex Systems', *SIAM Review*, **63**: 435–85.

Appendix A: Data documentation

All accessibility, hosting, and licensing information for HYPERBARD is summarized in Table 2.

Table 2. Accessibility, hosting, and licensing information for HYPERBARD.

Dataset Hosting Platform	Zenodo
Dataset Homepage	https://hyperbard.net
Dataset Tutorials	https://github.com/hyperbard/tutorials
Dataset DOI (original version)	10.5281/zenodo.6627159
Dataset DOI (latest version)	10.5281/zenodo.6627158
Dataset License	CC BY-NC 4.0
Code Hosting Platform	GitHub (maintenance), Zenodo (releases)
Code Repository	https://github.com/hyperbard/hyperbard
Code Documentation	https://hyperbard.readthedocs.io/en/latest/
Code DOI (original release)	10.5281/zenodo.6627161
Code DOI (latest release)	10.5281/zenodo.6627160
Code License	BSD 3-Clause

A.1 Datasheet

Our documentation follows the *Datasheets for Datasets* framework [8], omitting the questions referring specifically to data related to people.[2] For conciseness, unless otherwise indicated, the term *graph* refers to both *graphs* and *hypergraphs*.

A.1.1 Motivation

For what purpose was the dataset created? *Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.*

HYPERBARD was created to study the effects of modeling choices in the graph data curation process on the outputs produced by graph learning, graph mining, and network analysis algorithms.

There was no specific task in mind; rather, all classic graph learning, graph mining, and network analysis tasks were considered to be in scope. These tasks include, e.g. centrality ranking, outlier detection, clustering, similarity assessment, and standard statistical summarization, each for nodes, edges, and graphs, as well as variants of node classification, link prediction, or graph classification.

HYPERBARD was designed to fill a specific gap: Although there were myriad freely available graph datasets, to the best of our knowledge, none of them contained

- several different relational data representations,
- of the *same* underlying raw data,
- derived in a principled and well-documented manner,
- from each of several raw data instances belonging to a natural collection,
- where the raw data is intuitive and interpretable.

Who created the dataset (e.g. which team, research group) and on behalf of which entity (e.g. company, institution, organization)?

Corinna Coupette and Bastian Rieck created the dataset as part of their research.

Who funded the creation of the dataset? *If there is an associated grant, please provide the name of the grantor and the grant name and number.*

The creation of the dataset was indirectly funded by the institutions employing the dataset authors, i.e. the Max Planck Institute for Informatics (Corinna Coupette) and the Institute of AI for Health, Helmholtz Munich. There are no associated grants.

Any other comments?

None.

A.1.2 Composition

What do the instances that comprise the dataset represent (e.g. documents, photos, people, countries)? *Are there multiple types of instances (e.g. movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.*

Each instance represents a play attributed to William Shakespeare as a graph, and there are multiple different graph representations per play. In some graphs (i.e. hypergraphs and graphs derived from clique expansions of hypergraphs), nodes represent characters, and

(hyper)edges represent that characters were on stage at the same time in some part of the play. In other graphs (i.e. graphs derived from star expansions of hypergraphs), nodes represent characters or parts of a play, and an edge indicates that a character was on stage in that part of the play. The representations provided differ not only in their semantic mapping (what are the nodes and edges) but also in their granularity (what parts of the play are modeled as edges resp. nodes) and in their expressivity (what additional information is associated with nodes and edges); see Table 1 in the HYPERBARD paper.

How many instances are there in total (of each type, if appropriate)?

There are 37 plays in the raw data; 17 comedies, 10 historical plays, and 10 tragedies. Each play is represented as a graph in (at least) 18 different ways, for a total of 666 graph representations.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? *If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g. geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g. to cover a more diverse range of instances, because instances were withheld or unavailable).*

The dataset contains graph representations of all plays attributed to William Shakespeare by the Folger Shakespeare Library (see https://folgerpedia.folger.edu/William_Shakespeare%27s_plays), with the exception of lost plays and the comedy *The Two Noble Kingsmen*—a collaboration between Shakespeare and John Fletcher that is not currently provided in the TEI simple format by Folger Digital Texts.

What data does each instance consist of? *"Raw" data (e.g. unprocessed text or images) or features? In either case, please provide a description.*

Each instance, i.e. each of Shakespeare's plays, is represented by a set of files: one raw data file containing the text of the play as an XML encoded using the TEI Simple format, taken from Folger Digital Texts without modification, three CSV files containing preprocessed data, and 19 CSV files containing node lists and edge lists to construct different graph representations.

Consequently, dataset is distributed using the following folder structure:

- `rawdata`: contains 37 raw data XML files encoded in TEI simple.
- `data`: contains 3·37 preprocessed data files derived from files in `rawdata`.
- `graphdata`: contains 19·37 node and edge lists to construct graph representations from the files in `data`.
- `metadata`: contains `playtypes.csv`, mapping play identifiers to play types (comedy, history, or tragedy).

Python code to reproduce all graph representations and load them as *networkx* or *hypernetx* graphs is maintained in a GitHub repository (https://github.com/hyperbard/hyperbard), and code releases are archived via Zenodo (10.5281/zenodo.6627160).

Is there a label or target associated with each instance? *If so, please provide a description.*

There are labels corresponding to the type of play (one of {comedy, history, tragedy}), which could be used to partition the data for exploration, or as targets in classification tasks.

Is any information missing from individual instances? *If so, please provide a description, explaining why this information is missing (e.g. because it was unavailable). This does not include intentionally removed information, but might include, e.g. redacted text.*

There is no missing information.

Are relationships between individual instances made explicit (e.g. users' movie ratings, social network links)? *If so, please describe how these relationships are made explicit.*

When considering plays as instances, no relationships between individual instances are made explicit. When considering characters or parts of plays as instances, however, relationships between characters, or between characters and parts of plays are made explicit in the graph representations, exploiting the TEI Simple encoding of that data and the annotations provided in the XML attributes.

Are there recommended data splits (e.g. training, development/validation, testing)? *If so, please provide a description of these splits, explaining the rationale behind them.*

There are no recommended data splits for the current release.

Are there any errors, sources of noise, or redundancies in the dataset? *If so, please provide a description.*

The raw data contain some errors and redundancies in the XML encoding. Errors include redundant XML tags (e.g. doubly-wrapped tags), but also character entries or exits not explicitly annotated.

Redundancies result from the choice, made by the creators of Folger Digital Texts, to encode some information conveyed in the raw text also as attributes or separate XML tags (e.g. a character who speaks is encoded both as an attribute of the tag wrapping the speech and as an XML tag wrapping the name of the speaker).

There are two notable sources of noise affecting the preprocessed data and the graph data, both of which relate to our handling of stage directions—i.e. our processing of the XML attributes of `<stage>` tags in the raw data.

First, to determine which characters are on stage when a word is spoken, we primarily rely on the contents of `who` attributes in the `<stage>` tags of the raw data marked with `type="entry"` resp. `type="exit"`. The `who` attributes, however, are sometimes *semantically* incomplete, i.e. they may reflect Shakespeare's original stage directions accurately, but the original stage directions do not mention implied character movements (such as the exit of a side character or the exit of characters that died or fell unconscious at the end of a scene). To limit the impact of this noise source on our graph representations, we "flush" characters when a new scene starts (to handle missing exits) and ensure that the speaker is always on stage (to handle missing entries, some of which are also introduced by our character flushing policy).

Second, in our directed graph representations, where edges encode speaking and being spoken to, we equate being on stage while a word is spoken with hearing the word. Thus, we do not account for the impact of some stage directions concerning delivery, e.g. stage directions indicating that speech is inaudible for some or all other characters on stage, on the information flow our directed graph representations purport to capture. In the TEI simple encoding of our raw data, such stage directions are annotated with `type="delivery"`, but there is no indication of who can hear the words so delivered in the XML annotations. There are 2 200 XML tags annotated with `type="delivery"` (i.e. 60 delivery modifications per play on average). As modifications to delivery are sometimes crucial to drive the plot (e.g. by setting up misunderstandings), the impact of this noise source should not be underestimated, but it affects only our directed graph representations, which might be cautiously interpreted as "upper bounds" on the information flow between the characters on stage.

These sources of noise detailed above could likely be eliminated, to a large extent, by a more sophisticated parsing of the stage directions. This parsing could leverage, e.g. natural language processing methods to supplement the XML annotations. We plan to implement this improvement for a future dataset release.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g. websites, tweets, other datasets)? *If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e. including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g. licenses, fees) associated with any of the external resources that might apply to a dataset consumer? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.*

The dataset is self-contained. The raw data stem from Folger Digital Texts, maintained by the Folger Shakespeare Library and released under the CC BY-NC 3.0 Unported license, and they are redistributed without modifications as part of the HYPERBARD dataset. All other data are derived from the raw data, and the CC BY-NC 3.0 Unported license does not impose any additional restrictions. As part of our dataset maintenance (see below), we will regularly check Folger Digital Texts for modifications, and we will recompute and redistribute an updated HYPERBARD dataset under a versioned DOI whenever we detect changes.

Does the dataset contain data that might be considered confidential (e.g. data that is protected by legal privilege or by doctor–patient confidentiality, data that includes the content of individuals' nonpublic communications)? *If so, please provide a description.*

The dataset does not contain data that might be considered confidential.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? *If so, please describe why.*

The raw data, i.e. Shakespeare's plays, contain scenes that might be considered offensive, insulting, threatening, or otherwise anxiety-inducing from a contemporary perspective. For example, there is considerable controversy in the humanities around whether *The Taming of the Shrew* is misogynistic, and the main female protagonist's final speech on female submissiveness (Act V, SCENE 2, ll. 136–179) might cause discomfort to modern readers. Moreover, the corpus uses words that might be considered derogatory or offensive from a contemporary perspective. The preprocessed data, however, disassembles the original text, such that (offensive) play content is no longer immediately apparent when the data is viewed directly.

Any other comments?

The entire dataset takes up roughly 365 MB when uncompressed, and 30 MB when compressed.

A.1.3 Collection process

How was the data associated with each instance acquired? *Was the data directly observable (e.g. raw text, movie ratings), reported by subjects (e.g. survey responses), or indirectly inferred/derived from other data (e.g. part-of-speech tags, model-based guesses for age or language)?*

The raw data associated with each instance was acquired from Folger Digital Texts as XML files encoded in TEI Simple format. This format contains both raw text and structural, linguistic, and semantic annotations embedded in XML tags or XML attributes. Hence, it was partially directly observable (e.g. the raw text and its structure) and partially derived from other data (e.g. the XML tags and their attributes). The preprocessed data and the graph data were derived from the raw data.

If the data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? *If so, please describe how.*

To the extent that the raw data were indirectly inferred or derived from other data, validation was performed by the specialists from Folger Digital Texts. The preprocessed data and the graph data were validated by unit tests and manual inspection aided by visualizations (which also led us to discover the noise sources detailed above).

What mechanisms or procedures were used to collect the data (e.g. hardware apparatuses or sensors, manual human curation, software programs, software APIs)? *How were these mechanisms or procedures validated?*

The raw data was bulk downloaded in TEI Simple format as a ZIP archive from the Folger Digital Texts downloads section, and Folger Digital Texts compiled the raw data through computer-assisted manual curation. The bulk download was checked manually to ensure that the extracted archive contained one XML file per play, as expected. The code creating the preprocessed data from the raw data and the graph representations from the preprocessed data is almost completely unit tested.

If the dataset is a sample from a larger set, what was the sampling strategy (e.g. deterministic, probabilistic with specific sampling probabilities)?

The data is not a sample from a larger set.

Who was involved in the data collection process (e.g. students, crowdworkers, contractors) and how were they compensated (e.g. how much were crowdworkers paid)?

Only Corinna Coupette and Bastian Rieck, the dataset authors, were involved in the data collection process.

Over what timeframe was the data collected? *Does this timeframe match the creation timeframe of the data associated with the instances (e.g. recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created.*

The raw data was collected through one download call to https://shakespeare.folger.edu/downloads/teisimple/shakespeares-works_TEIsimple_FolgerShakespeare.zip in June 2022, and the preprocessed data and the graph data were derived from the raw data by running a code pipeline, also in June 2022. This timeframe does not match the creation timeframe of the raw data, which, though internal to the Folger Shakespeare Library, spans at least several months in 2020. It also does not match the creation timeframe of Shakespeare's plays, which spans several decades in the 16th and 17th centuries.

Were any ethical review processes conducted (e.g. by an institutional review board)? *If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.*

No ethical review processes were conducted.

Any other comments?

None.

A.1.4 Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g. discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? *If so, please provide a description. If not, you may skip the remaining questions in this section.*

Our data preprocessing consists of two steps.

1) Transform raw XML data into preprocessed CSV data (`rawdata` → `data`).
   Script: `run_preprocessing.py`
   a) Extract the cast list from the TEI Simple XML and store it as a CSV. (This is technically unnecessary to generate our graph representations, but it gives a convenient overview of the characters occurring in the play.)

Function: `get_cast_df`
Artifact: `data/{play}.cast.csv`

b) Parse the TEI Simple XML into a table containing one row per descendant of the TEI Simple <body> tag, and the tag names and XML attributes of all XML tags of interest (eliminating redundant XML elements), plus the text content of all XML tags that are leaves, as columns. Annotate the result with information on the act and scene in which the tag occurs, the characters on stage when the tag occurs, and the speaker(s), if any.
Function `get_raw_xml_df`
Artifact: `data/{play}.raw.csv`

c) Transform the artifact from the previous step into a table with one row per setting on stage, where a setting is a stretch of the play without changes to the speaker or to the group of characters on stage, and information on the setting as well as the number of lines and tokens spoken in that setting as columns.
Artifact: `data/{play}.agg.csv`

2) Transform preprocessed CSV data into node and edge CSV files for graph construction (`data` → `graphdata`).
The artifacts resulting from this step are generally labeled `{play}_{semantic mapping}_{granularity}_{expressivity}.{list type}.csv`, omitting the expressivity (and granularity) components in node lists if all different graph representations with a given semantic mapping (and granularity) use the same set of nodes.

a) Create node lists and edge lists for different graph representations in CSV format from `data/{play}.agg.csv` artifacts.
Script: `create_graph_representations.py`
Artifacts:
- `graphdata/{play}_ce-group-mw.edges.csv`
- `graphdata/{play}_ce-group-w.edges.csv`
- `graphdata/{play}_ce-scene-mw.edges.csv`
- `graphdata/{play}_ce-scene-w.edges.csv`
- `graphdata/{play}_ce.nodes.csv`
- `graphdata/{play}_se-group-w.edges.csv`
- `graphdata/{play}_se-group.nodes.csv`
- `graphdata/{play}_se-scene-w.edges.csv`
- `graphdata/{play}_se-scene.nodes.csv`
- `graphdata/{play}_se-speech-mwd.edges.csv`
- `graphdata/{play}_se-speech-wd.edges.csv`
- `graphdata/{play}_se-speech.nodes.csv`

b) Create node lists and edge lists for different hypergraph representations in CSV format from `data/{play}.agg.csv` artifacts.
Script: `create_hypergraph_representations.py`
Artifacts:
- `graphdata/{play}_hg-group-mw.edges.csv`
- `graphdata/{play}_hg-group-mw.node-weights.csv`
- `graphdata/{play}_hg-scene-mw.edges.csv`
- `graphdata/{play}_hg-scene-mw.node-weights.csv`
- `graphdata/{play}_hg-speech-mwd.edges.csv`
- `graphdata/{play}_hg-speech-wd.edges.csv`
- `graphdata/{play}_hg.nodes.csv`

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g. to support unanticipated future uses)? *If so, please provide a link or other access point to the "raw" data.*

The raw data was saved, and it is distributed along with the preprocessed data in the dataset available from Zenodo under a versioned DOI: 10.5281/zenodo.6627158.

Is the software that was used to preprocess/clean/label the data available? *If so, please provide a link or other access point.*

The software used to transform the raw data into the preprocessed data, and the preprocessed data into the graph data representations, is available on GitHub in the following repository: https://github.com/hyperbard/hyperbard.

All code releases are also available on Zenodo under a versioned DOI: 10.5281/zenodo.6627160.

Any other comments?

All data preprocessing can be completed in a couple of minutes even on older commodity hardware. We used

a 2016 MacBook Pro with a 2.9 GHz Quad-Core Intel Core i7 processor and 16 GB RAM.

A.1.5 Uses

Has the dataset been used for any tasks already? *If so, please provide a description.*

In the paper introducing HYPERBARD, the dataset has been used to demonstrate the differences between rankings of characters by degree that result from different modeling choices made when transforming raw data into graphs.

Is there a repository that links to any or all papers or systems that use the dataset? *If so, please provide a link or other access point.*

Papers or systems known to use dataset will be collected on https://hyperbard.net and on GitHub.

What (other) tasks could the dataset be used for?

HYPERBARD was designed for inquiries into the stability of algorithmic results under different reasonable representations of the underlying raw data, i.e. to enable *representation robustness checks* for graph learning, graph mining, and network analysis methods. In this role, it could generally be used for all graph learning, graph mining, and network analysis tasks identified as *in scope* in the motivation section.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? *For example, is there anything that a dataset consumer might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g. stereotyping, quality of service issues) or other risks or harms (e.g. legal risks, financial harms)? If so, please provide a description. Is there anything a dataset consumer could do to mitigate these risks or harms?*

The quality and expressivity of the dataset is limited by the quality and expressivity of Folger Digital Texts encoded using the TEI Simple format, which could restrict usage in the digital humanities, e.g. when they are interested in the minute details of character interactions described in stage directions.

HYPERBARD contains relational data representations of Shakespeare's plays, which were written more than four centuries ago. Hence, there are no risks or harms associated with the dataset beyond the risks or harms also associated with the ongoing study of Shakespeare's works in the humanities, and the risks or harms associated with the decontextualization or overinterpretation of any dataset.

At https://hyperbard.net and on GitHub, we keep a continuously-updated list of all known dataset limitations for dataset consumers to review when deciding whether HYPERBARD is appropriate for their use case.

Are there tasks for which the dataset should not be used? *If so, please provide a description.*

Outside *representation robustness checks*, HYPERBARD should not be used in tasks that have no reasonable semantic interpretation in the domain of the raw data.

Any other comments?

None.

A.1.6 Distribution

Will the dataset be distributed to third parties outside of the entity (e.g. company, institution, organization) on behalf of which the dataset was created? *If so, please provide a description.*

The dataset was not created on behalf of any entity, and it will be distributed freely.

How will the dataset will be distributed (e.g. tarball on website, API, GitHub)? *Does the dataset have a digital object identifier (DOI)?*

The dataset will be distributed as a ZIP archive via Zenodo, based on code hosted on GitHub. Each dataset version and each code release will have a versioned DOI, generated automatically by Zenodo. See also Table 2.

When will the dataset be distributed?

The dataset will be distributed when the paper introducing it is submitted.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? *If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.*

The dataset will be distributed under a CC BY-NC 4.0 license, according to which others are free to

- *share*, i.e. copy and redistribute, and
- *adapt*, i.e. remix, transform, and build on the material, provided they
- *give attribution*, i.e. give appropriate credit, provide a link to the license, and indicate if changes were made,
- do *not* use the material for *commercial purposes*, and

- *add no restrictions* limiting others in doing anything the license permits.

The code constructing the dataset will be distributed under a permissive BSD 3-Clause license.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? *If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.*

The Folger Shakespeare Library has released the source of our raw data, Folger Digital Texts, under the CC BY-NC 3.0 Unported license, which has essentially the same usage conditions as our CC BY-NC 4.0 license.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? *If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.*

No export controls or other regulatory restrictions apply.

Any other comments?

None.

A.1.7 Maintenance

Who will be supporting/hosting/maintaining the dataset?

Corinna Coupette and Bastian Rieck will be supporting, hosting, and maintaining the dataset.

How can the owner/curator/manager of the dataset be contacted (e.g. email address)?

In the interest of transparency, the preferred method to contact the dataset maintainers is by opening GitHub issues at https://github.com/hyperbard/hyperbard. Alternatively, the dataset maintainers can be reached by email to info@hyperbard.net.

Is there an erratum? *If so, please provide a link or other access point.*

Errata will be documented at https://hyperbard.net and on GitHub.

Will the dataset be updated (e.g. to correct labeling errors, add new instances, delete instances)? *If so, please describe how often, by whom, and how updates will be communicated to dataset consumers (e.g. mailing list, GitHub)?*

The dataset will be updated as needed, and updates will be labeled using *semantic versioning*.

- A *patch version* (e.g. 0.0.1 → 0.0.2) is a recomputation of the latest dataset version following a non-breaking change in the underlying raw data.
- A *minor version* (e.g. 0.0.1 → 0.2.0) is an update of the latest dataset version that increases the expressivity of existing representations while maintaining all of their previously present features.
- Any other update is a *major version* (e.g. 0.0.1 → 1.0.0). This includes, e.g. responses to breaking changes in the underlying source data, additions of new representations, and changes to existing representations that might break dataset consumer code.

Patch versions will be created automatically using GitHub actions. Minor versions and major versions will be created by the dataset maintainers, potentially accepting pull requests or implementing feature requests filed via at https://github.com/hyperbard/hyperbard.

New releases will be communicated at https://hyperbard.net and on GitHub, and they will be available for download under a versioned DOI on Zenodo, with 10.5281/zenodo.6627158 always resolving to the latest release.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g. were the individuals in question told that their data would be retained for a fixed period of time and then deleted)? *If so, please describe these limits and explain how they will be enforced.*

There are no data retention limits.

Will older versions of the dataset continue to be supported/hosted/maintained? *If so, please describe how. If not, please describe how its obsolescence will be communicated to dataset consumers.*

Older versions of the dataset will remain hosted on Zenodo, with the relevant version of the code needed to reproduce them available in an associated GitHub release, also archived on Zenodo.

There will be basic support for older versions of the dataset, and as HYPERBARD is derived from century-old literary works, dataset maintenance amounts to dataset updates (see the paragraph on dataset updates).

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? *If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to dataset consumers? If so, please provide a description.*

Others can extend, augment, build on, and contribute to the dataset through the engagement mechanisms provided by GitHub.

See also https://github.com/hyperbard/hyperbard/blob/main/CONTRIBUTING.md.

Extensions, augmentations, and contributions provided via pull requests will be validated and verified by the dataset maintainers in a regular code and data review process, while changes made in independent forks will not be checked.

Contributions integrated with the HYPERBARD code repository will be visible on GitHub, and they trigger new dataset releases, in which contributors will be specifically acknowledged.

Any other comments?

None.

A.2 Hosting, license, and maintenance plan

For hosting and licensing information, see Table 2 and Section A.1.6. For the maintenance plan, see Section A.1.7.

A.3 Author responsibility statement

The dataset authors, Corinna Coupette and Bastian Rieck, bear all responsibility in case of violation of rights, etc., and they confirm that the data is released under the CC BY-NC 4.0 license, and that the code is released under the BSD 3-Clause license.

Appendix B: Usage documentation

The HYPERBARD dataset is distributed in four folders: `rawdata`, `data`, `graphdata`, and `metadata`. See Section A.1.2 for more details on the composition of the dataset. The dataset can be reproduced by cloning the GitHub repository and running `make` (this will also generate most figures included in the HYPERBARD paper).

In addition to the written documentation, we provide Jupyter notebook tutorials for interactive data exploration. The tutorials are hosted on GitHub at https://github.com/hyperbard/tutorials, and they can be run both locally and in a Binder, i.e. a fully configured remote environment accessible through the browser without any local setup. Launching the Binder usually takes around thirty seconds.

In the following, we explain the structure of the files in HYPERBARD's folders and detail how these files can be read. All file examples are taken from *Romeo and Juliet*, and for CSV files, all columns are described in alphabetical order.

B.1 `rawdata`

This folder contains XML files encoded in TEI Simple as provided by Folger Digital Texts. These files can be read with any XML parser, such as the parser from the `beautifulsoup4` library in Python. All file names follow the pattern `{play}_TEIsimple_FolgerShakespeare.xml`.

The XML encoding is designed to meet the needs of the (digital) humanities, and hence, it is very detailed and fine-grained. For example, every word, whitespace character, and punctuation mark is contained in its own tag.

The encoding practices followed by Folger Digital Texts are described in the `<encodingDesc>` tag of each text. To summarize:

- The major goal of the TEI Simple encoding is to achieve interoperability with a large corpus of early modern texts derived from the Early English Books Text Creation Partnership transcriptions (i.e. it is different from *our* goal).
- The encoding is completely faithful to the readings, orthography, and punctuation of the source texts (i.e. the Shakespeare texts edited by Barbara Mowat and Paul Werstine at Folger Shakespeare Library).
- All `xml:ids` are corpuswide identifiers (i.e. they are unique across all our plays, too).
- Words, spaces, and punctuation characters are numbered sequentially within each play, incremented by 10 (XML attribute: `n`).
- Most other elements begin with an element-specific prefix, followed by a reference to the Folger Through Line Number, a sequential numbering of the numbered lines in the text. (Details omitted.)
- Spoken words are linguistically annotated with a lemma and POS tag.

Running the script `compute_rawdata_xml_statistics.py` in the HYPERBARD GitHub repository, which computes basic XML tag, path, and attribute statistics for the entire corpus and writes the results to the `metadata` folder as CSV files, provides some intuition regarding the structure of the raw data. This script also pulls the descriptions of all tags from the current TEI specification. For more information on the TEI Simple format, which has been integrated with the main TEI specification, see https://github.com/TEIC/TEI-Simple.

Example:

```
...
<sp xml:id="sp-0015" who="#SERVANTS.CAPULET.Sampson_Rom">
<speaker xml:id="spk-0015">
<w xml:id="fs-rom-0002610">SAMPSON</w>
</speaker>
<p xml:id="p-0015">
<lb xml:id="ftln-0015" n="1.1.1"/>
<w xml:id="fs-rom-0002620" n="1.1.1" lemma="Gregory" ana="#n1-nn">Gregory</w>
<pc xml:id="fs-rom-0002630" n="1.1.1">,</pc>
<c> </c>
<w xml:id="fs-rom-0002650" n="1.1.1" lemma="on" ana="#acp-p">on</w>
<c> </c>
<w xml:id="fs-rom-0002670" n="1.1.1" lemma="my" ana="#po">my</w>
<c> </c>
<w xml:id="fs-rom-0002690" n="1.1.1" lemma="word" ana="#n1">word</w>
<c> </c>
<w xml:id="fs-rom-0002710" n="1.1.1" lemma="we|will" ana="#pns|vmb">we'll</w>
<c> </c>
<w xml:id="fs-rom-0002730" n="1.1.1" lemma="not" ana="#xx">not</w>
<c> </c>
<w xml:id="fs-rom-0002750" n="1.1.1" lemma="carry" ana="#vvi">carry</w>
<c> </c>
<w xml:id="fs-rom-0002770" n="1.1.1" lemma="coal" ana="#n2">coals</w>
<pc xml:id="fs-rom-0002780" n="1.1.1">.</pc>
</p>
</sp>
...
```

B.2 `data`

This folder contains CSV files, which can be read with any CSV parser, such as the parser from the `pandas` library in Python.

There are three types of files:

`{play}.cast.csv` files, `{play}.raw.csv` files, and `{play}.agg.csv` files.

B.2.1 `{play}.cast.csv`

A `{play}.cast.csv` file contains the XML identifiers and attributes of all `<castItem>` tags found in a `{play}_TEIsimple_FolgerShakespeare.xml` file. It gives an overview of the characters occurring in a play, and it can be used to count the number of characters (including characters that do not speak) or to build a hierarchy of characters and character groups.

Rows correspond to characters or character groups.

Columns in alphabetical order:

- `corresp`: group (i.e. another cast item) to which a given cast item belongs, if any (XML attribute abbreviating "corresponds").

  Type: String or NaN (if the cast item does not belong to any other cast item).
- `xml:id`: unique identifier of the cast member.

  Type: String.

Note that the data in each of these columns does *not* start with a # sign. This contrasts with *references* to the `xml:ids` in the attributes of other XML tags in the raw data XML files, which *do* start with a # sign (to indicate the referencing).

Example:

```
xml:id,corresp
ATTENDANTS.PRINCE_Rom,ATTENDANTS_Rom
ATTENDANTS_Rom,
Apothecary_Rom,
Benvolio_Rom,
Boy_Rom,
...
```

B.2.2 `{play}.raw.csv`

A {play}.raw.csv file contains the descendants of the `<body>` tag found in a `{play}_TEIsimple_FolgerShakespeare.xml` file, with redundancies resulting from the encoding format eliminated, and additional information to build graph representations annotated. It provides a *disaggregated* tabular overview of the information underlying our graph representations, and it serves as the basis of its corresponding `{play}.agg.csv` file.

Rows correspond to instances of XML tags.

Columns in alphabetical order:

- `act`: Derived attribute. The number of the act in which the tag occurs. An integer in [5] for all tags in the main part of the play. 0 for tags occurring before the first act (e.g., in a prologue or an induction), 6 for tags occurring after the fifth act (e.g., in an epilogue).
  Type: Non-negative integer.
- `ana`: Original attribute. If the tag wraps a spoken word, the POS tag of that word (XML attribute abbreviating "analysis").
  Type: String or NaN (if the tag does not wrap a spoken word).
- `lemma`: Original attribute. If the tag wraps a spoken word, the lemma of that word.
  Type: String or NaN (if the tag does not wrap a spoken word).
- `n`: Original attribute. A label for the element, not necessarily unique.
  Type: String, positive integer (for `<div>` tags representing acts or scenes), or NaN (e.g., for `<c>` tags wrapping whitespace characters).
- `onstage`: Derived attribute. Whitespace-separated list of characters on stage when the tag occurs.
  Type: String or NaN.
- `part`: Original attribute. Rare and not of interest for graph building.
  Type: String or NaN.
- `prev`: Original attribute. Rare and not of interest for graph building.
  Type: String or NaN.
- `rendition`: Original attribute. Rare and not of interest for graph building.
  Type: String or NaN.
- `scene`: Derived attribute. The number of the scene in which the tag occurs. 0 if the tag does not occur in a scene.

Type: Non-negative integer.
- `speaker`: Derived attribute. Whitespace-separated list of characters who are speaking when a tag occurs. Note that several characters can speak at the same time, although the overwhelming majority of speech in the corpus is uttered by only one speaker.
  Type: String or NaN.
- `stagegroup_raw`: Derived attribute. Number stating how many changes in the set of characters on stage we have already witnessed when a tag occurs (i.e. the same set of characters can occur in different stage groups). Relevant for sorting and aggregation.
  Type: Non-negative integer.
- `tag`: Original entity. The name of the XML tag to which the row corresponds.
  Type: String.
- `text`: Original text content.
  Type: String or NaN (if a tag is not a leaf in the XML tree).
- `type`: Original attribute. Used to give details on and <stage> tags, e.g., distinguish between acts and scenes, and mark stage directions as, e.g., character entry or exit.
  Type: String or NaN.
- `who`: Original attribute giving information on characters who act, transformed into a set. Will become whitespace-separated list in future releases.
  Type: Set of strings or NaN.
- `xml: id`: Original XML identifier. Note that instances of some XML tags, including and <c> tags, do not have XML identifiers.
  Type: String or NaN.

Example:

```
tag,type,n,text,xml:id,who,lemma,ana,
part,rendition,prev,act,scene,
onstage,stagegroup_raw,speaker
```

```
...
sp,,,,sp-0015,{'#SERVANTS.CAPULET.Sampson_Rom'},,,,,,1,1,#SERVANTS.
    CAPULET.Gregory_Rom #SERVANTS.CAPULET.Sampson_Rom,3,#SERVANTS.
    CAPULET.Sampson_Rom
p,,,,p-0015,,,,,,,,1,1,#SERVANTS.CAPULET.Gregory_Rom #SERVANTS.CAPULET.
    Sampson_Rom,3,
lb,,1.1.1,,ftln-0015,,,,,,,1,1,#SERVANTS.CAPULET.Gregory_Rom #SERVANTS.
    CAPULET.Sampson_Rom,3,#SERVANTS.CAPULET.Sampson_Rom
w,,1.1.1,Gregory,fs-rom-0002620,,Gregory,#n1-nn,,,,1,1,#SERVANTS.CAPULET
    .Gregory_Rom #SERVANTS.CAPULET.Sampson_Rom,3,#SERVANTS.CAPULET.
    Sampson_Rom
pc,,1.1.1,",",fs-rom-0002630,,,,,,,1,1,#SERVANTS.CAPULET.Gregory_Rom #
    SERVANTS.CAPULET.Sampson_Rom,3,#SERVANTS.CAPULET.Sampson_Rom
c,,, ,,,,,,,,,1,1,#SERVANTS.CAPULET.Gregory_Rom #SERVANTS.CAPULET.
    Sampson_Rom,3,
w,,1.1.1,on,fs-rom-0002650,,on,#acp-p,,,,1,1,#SERVANTS.CAPULET.
    Gregory_Rom #SERVANTS.CAPULET.Sampson_Rom,3,#SERVANTS.CAPULET.
    Sampson_Rom
c,,, ,,,,,,,,,1,1,#SERVANTS.CAPULET.Gregory_Rom #SERVANTS.CAPULET.
    Sampson_Rom,3,
...
```

B.2.3 {play}.agg.csv

A {play}.agg.csv file contains a condensed and filtered view of its corresponding {play}.raw.csv file, focusing only on spoken words. It provides an *aggregated* tabular overview of the information underlying our graph representations, and it serves as the basis of all files in the `graphdata` folder. In contrast to the {play}.raw.csv file, which contains some original attributes, {play}.agg.csv contains only derived attributes.

Rows correspond to *settings* (or *speech acts*), i.e. maximal sequences of words in which neither the speaker(s) nor the group of characters on stage change.

Columns in alphabetical order:

- `act`: The same as `act` in {play}.raw.csv.
- `n_lines`: The number of lines spoken in a setting.
  Type: Positive integer.
- `n_tokens`: The number of tokens spoken in a setting.
  Type: Positive integer.
- `onstage`: The same as `onstage` in {play}.raw.csv.
- `scene`: The same as `scene` in {play}.raw.csv.
- `setting`: Number stating how many changes in the tuple (set of characters on stage, speaker) we have seen when the words summarized in this row occur, plus 1 (for consistency with the numbering in `stagegroup`).
  Type: Positive integer.
- `speaker`: The same as `speaker` in {play}.raw.csv.
- `stagegroup`: The contents of the `stagegroup_raw` column, renumbered to be consecutive in {play}.agg.csv, starting with 1.
  Type: Positive integer.
- `stagegroup_raw`: The same as `stagegroup_raw` in {play}.raw.csv.

Example:

```
act,scene,stagegroup,stagegroup_raw,
setting,onstage,speaker,n_lines,
n_tokens
0,0,1,1,1,#Chorus_Rom,
#Chorus_Rom,14,106
1,1,2,3,2,#SERVANTS.CAPULET.
Gregory_Rom #SERVANTS.CAPULET.Sampson_
Rom,#SERVANTS.CAPULET.Sampson_Rom,1,8
1,1,2,3,3,#SERVANTS.CAPULET.Gregory_
Rom #SERVANTS.CAPULET.Sampson_
Rom,#SERVANTS.CAPULET.Gregory_
Rom,1,7
1,1,2,3,4,#SERVANTS.CAPULET.Gregory_
Rom #SERVANTS.CAPULET.Sampson_
Rom,#SERVANTS.CAPULET.Sampson_
Rom,1,9
```

`1,1,2,3,5,#SERVANTS.CAPULET.Gregory_Rom #SERVANTS.CAPULET.Sampson_Rom,#SERVANTS.CAPULET.Gregory_Rom,2,10`

B.3 `graphdata`

This folder contains CSV files, which can be read with any CSV parser, such as the parser from the `pandas` library in Python.

For each play, the folder holds all files needed to generate the representations listed in Table 1, i.e.:

- Files to construct *clique expansions* (`ce`, i.e. character co-occurrence networks):
  - `{play}_ce-group-mw.edges.csv`:
    Weighted multi-edges for clique expansions aggregated at the stage group level.
    Use to generate ce-group-{mb, mw} representations.
  - `{play}_ce-group-w.edges.csv`:
    Count-weighted edges for clique expansions aggregated at the stage group level.
    Use to generate ce-group-b representations (or ce-group-w representations for easier plotting of ce-group-mb representations if the edge order does not matter).
  - `{play}_ce-scene-mw.edges.csv`:
    Weighted multi-edges for clique expansions aggregated at the scene level.
    Use to generate ce-scene-{mb, mw} representations.
  - `{play}_ce-scene-w.edges.csv`:
    Count-weighted edges for clique expansions aggregated at the scene level.
    Use to generate ce-scene-b representations (or ce-scene-w representations for easier plotting of ce-scene-mb representations if the edge order does not matter).
  - `{play}_ce.nodes.csv`:
    Nodes for all clique expansions.
    Use to generate all ce-$*$ representations.
- Files to construct *star expansions* (`se`, i.e. bipartite graphs with characters and text units as node sets):
  - `{play}_se-group-w.edges.csv`:
    Edges for star expansions aggregated at the stage group level.
    Use to generate se-group-{b, w} representations.
  - `{play}_se-group.nodes.csv`:
    Nodes for star expansions aggregated at the stage group level.
    Use to generate se-group-{b, w} representations.
  - `{play}_se-scene-w.edges.csv`:
    Edges for star expansions aggregated at the scene level.
    Use to generate se-scene-{b, w} representations.
  - `{play}_se-scene.nodes.csv`:
    Nodes for star expansions aggregated at the scene level. The character nodes are the same as for `{play}_se-group.nodes.csv`, but the text unit nodes differ.
    Use to generate se-scene-{b, w} representations.
  - `{play}_se-speech-mwd.edges.csv`:
    Directed multi-edges for star expansions aggregated at the speech act level. Multi-edges can occur because there exists one edge per speech act, but text unit nodes are resolved at the stage group level, and one stage group can contain several speech acts.
    Use to generate the se-speech-mwd representation.
  - `{play}_se-speech-wd.edges.csv`:
    Directed edges for star expansions aggregated at the speech act level, with multi-edges aggregated into edge weights.
    Use to generate the se-speech-wd representation.
  - `{play}_se-speech.nodes.csv`:
    Nodes for star expansions aggregated at the speech act level. The same as `{play}_se-group.nodes.csv`; provided separately to facilitate the matching between node and edge files.
    Use to generate se-speech-{wd, mwd} representations.
- Files to construct *hypergraphs* (`hg`, i.e. generalized graph representations allowing edges with cardinalities in $\mathbb{N}$):
  - `{play}_hg-group-mw.edges.csv`:
    Edges for hypergraph representations resolved at the stage group level.
    Use to generate hg-group-{mb, mw} representations.
  - `{play}_hg-group-mw.node-weights.csv`:
    Edge-specific node weights for hypergraph representations resolved at the stage group level.
    Use to generate hg-group-{mb, mw} representations with edge-specific node weights.
  - `{play}_hg-scene-mw.edges.csv`:
    Edges for hypergraph representations resolved at the scene level.
    Use to generate hg-scene-{mb, mw} representations.
  - `{play}_hg-scene-mw.node-weights.csv`:

Edge-specific node weights for hypergraph representations resolved at the scene level.
Use to generate hg-scene-{mb, mw} representations with edge-specific node weights.

- `{play}_hg-speech-mwd.edges.csv`:
  Directed, weighted multi-edges for hypergraph representations resolved at the speech act level, where both the source and the target can contain multiple nodes.
  Use to generate the hg-speech-mwd representation.
- `{play}_hg-speech-wd.edges.csv`:
  Directed, weighted edges for hypergraph representations resolved at the speech act level, where both the source and the target can contain multiple nodes, with multi-edges aggregated into edge weights
  Use to generate the hg-speech-wd representation.
- `{play}_hg.nodes.csv`:
  Nodes for all hypergraph representations. Technically redundant because hyperedges can have cardinality 1, too, such that all nodes can be derived from the edge lists. Provided with global node weights for convenience.
  Use to generate all hg-$*$ representations.

The rows in each file represent either nodes or edges.

The columns in the individual files differ depending on the *semantic mapping*, the *granularity*, and the *expressivity* of the file contents, all of which are expressed in the file name (cf. Table 1), but the column semantics should be intuitive in light of the details on the `{play}.agg.csv` file columns given above. Note the following conventions for column names in edge lists:

- For clique and star expansions, if the graph is undirected, the nodes are called `node1` and `node2`, and if the graph is directed, the nodes are called `source` and `target`.
- If edges are count-weighted, the weight column is called `count`, otherwise, the columns `n_tokens` and `n_lines` can both serve as edge weights.
- For multi-edges in clique and star expansions, the column `edge_index` ensures that there are no duplicate rows. In hypergraphs, this is ensured by the `setting` column.

Finally, when working with the edge lists, please refer to the *expressivity* column in Table 1 to check whether the edge ordering in any particular file is intrinsically meaningful.

Examples:

- Nodes for clique expansions:

```
node
#ATTENDANTS.PRINCE_Rom
#ATTENDANTS_Rom
#Apothecary_Rom
#Benvolio_Rom
#Boy_Rom
    ...
```

- Edges for clique expansions (here: ce-group-mw):

```
node1,node2,key,act,scene,stagegroup,
n_tokens,n_lines,edge_index

#SERVANTS.CAPULET.Gregory_
Rom,#SERVANTS.CAPULET.Sampson_Rom,
0,1,1,2,254,33,2
  #SERVANTS.CAPULET.Gregory_
  Rom,#SERVANTS.CAPULET.Sampson_Rom,
  1,1,1,3,149,25,3
  #SERVANTS.CAPULET.Gregory_
  Rom,#SERVANTS.MONTAGUE.1_Rom,
  0,1,1,3,149,25,3
  #SERVANTS.CAPULET.Gregory_
  Rom,#SERVANTS.MONTAGUE.Abram_Rom,
  0,1,1,3,149,25,3
  #SERVANTS.CAPULET.Sampson_
  Rom,#SERVANTS.MONTAGUE.1_Rom,
  0,1,1,3,149,25,3
  ...
```

- Nodes for star expansions (here: se-group):

```
node,node_type
#ATTENDANTS.PRINCE_Rom,character
#ATTENDANTS_Rom,character
#Apothecary_Rom,character
...
0.00.0001,text_unit
1.01.0002,text_unit
1.01.0003,text_unit
...
```

- Edges for star expansions (here: se-speech-mwd):

```
source,target,key,n_lines,n_tokens,
edge_index,edge_type
#Chorus_Rom,0.00.0001,0,14,106,1,
active
#SERVANTS.CAPULET.Sampson_Rom,
1.01.0002,0,1,8,2,active
1.01.0002,#SERVANTS.CAPULET.
Gregory_
Rom,0,1,8,2,passive
#SERVANTS.CAPULET.Gregory_Rom,
1.01.0002,0,1,7,3,active
1.01.0002,#SERVANTS.CAPULET.Sampson_
Rom,0,1,7,3,passive
...
```

- Nodes for hypergraphs:
```
node,n_tokens_onstage,n_tokens_s-
peaker,n_lines_onstage,
n_lines_speaker
#ATTENDANTS.PRINCE_Rom,1147,0,150,0
#ATTENDANTS_Rom,905,0,121,0
#Apothecary_Rom,224,53,29,7
#Benvolio_Rom,5671,1160,771,161
#Boy_Rom,905,0,121,0
...
```
- Edge-specific node weights for hypergraphs (here: hg-scene-mw):
```
act,scene,node,n_tokens_speaker,
n_lines_speaker,n_tokens_onstage,
n_lines_onstage
0,0,#Chorus_Rom,106,14,106,14
1,1,#Benvolio_Rom,376,52,1403,189
1,1,#CITIZENS_Rom,16,2,237,32
1,1,#Capulet_Rom,26,3,221,30
1,1,#LadyCapulet_Rom,10,2,221,30
...
```
- Edges for hypergraphs (here: hg-speech-mwd):
```
act,scene,stagegroup,setting,
speaker,onstage,n_tokens,n_lines
0,0,1,1,#Chorus_Rom,
#Chorus_Rom,106,14
1,1,2,2,#SERVANTS.CAPULET.Sampson_
Rom,#SERVANTS.CAPULET.Gregory_Rom
#SERVANTS.CAPULET.Sampson_Rom,8,1
1,1,2,3,#SERVANTS.CAPULET.Gregory_
Rom,#SERVANTS.CAPULET.Gregory_Rom
#SERVANTS.CAPULET.Sampson_Rom,7,1
1,1,2,4,#SERVANTS.CAPULET.Sampson_
Rom,#SERVANTS.CAPULET.Gregory_Rom
#SERVANTS.CAPULET.Sampson_Rom,9,1
...
```

B.4 `metadata`

This folder currently contains exactly one CSV file, which maps play identifiers to play types. The file can be read with any CSV parser, such as the parser from the `pandas` library in Python, but since its provenance is documented as a comment at the start of the file, the # character needs to be passed to the parser as a comment character.

Rows correspond to plays.

Columns in alphabetical order:

- `play_name`: The name of the play, as used to fill the {play} placeholder in all play-specific file names.
  Type: String.
- `play_type`: The type of the play. One of {comedy, history, tragedy}.
  Type: String.

Appendix C: Contribution documentation

In the following, for context and accessibility, we summarize the story of the full play [5] as well as its two main themes, the dataset and the community critique.

C.1 The story

Induction, Scene I. Confronted by Reviewer, Authors explain their first contribution. *Act I, Scene I.* Creature gets drawn into the Community by Senior Researcher and Tutor. Welcomed by Professor, they sign their PhD contract. *Act I, Scene II.* Creature quarrels with their new role. They meet Colleague, their office mate, and three Deadlines, introduced by Professor. They submit to First Deadline. *Act I, Scene III.* Creature dreams of Hyperbard, a faun caring for raw data, and Graph, one of their spirits. They discuss how to obtain insights from raw data via transformations, and that each raw data point permits several relational representations. *Act II, Scene I.* Creature converses with Colleague, Professor, and Senior Researcher over lunch. They ask Colleague about the provenance of graph data used in the Community, and they learn about graph data repositories. *Act II, Scene II.* Creature revisits their dream. They identify semantic mapping, granularity, and expressivity as the dimensions in which several graph representations of the same raw data may differ. *Act II, Scene III.* Creature secretly observes Colleague as they mechanically prepare a graph dataset and produce a datasheet in the process. *Act II, Scene IV.* Confused and depressed by the practices they witness in the Community, Creature attempts suicide. *Act II, Scene V.* Outside the Community, Creature is cared for by Graph and Hyperbard. Together, the three of them develop the graph and hypergraph representations of Shakespeare's plays included in the Hyperbard dataset. *Act III, Scene I.* Creature gets haunted by the three Deadlines, who remind them of their ignoble academic incentives. They contemplate quitting their PhD. *Act IV, Scene I.* Accompanied by Graph and Hyperbard, Creature returns to the Community. They meet Professor, who calls Creature into their office and demands that Hyperbard leaves. *Act IV, Scene II.* From Professor, Creature learns that their paper got accepted. *Act IV, Scene III.* In the absence of Creature, Hyperbard and Graph try to convey their message that representations matter to Colleague. Professor and Creature return, and Professor orders Colleague to eliminate Hyperbard. *Act V, Scene I.* Having cremated Hyperbard, Colleague pours their ashes onto the graph dataset prepared earlier. Graph mourns the death of their sovereign and sketches its implications. *Act V, Scene II.* Creature wrestles with their

experience in the Community. Instead of leaving in silence, they decide to tell their own story.

C.2 The dataset

The HYPERBARD dataset comprises 666 graphs and hypergraphs: 18 relational representations for each of 37 plays by William Shakespeare (Fig. 1). From the TEI Simple XMLs provided by Folger Digital Texts [14], for each play, we derive 6 hypergraphs, 6 clique expansions (i.e. interaction graphs), and 6 star expansions (i.e. bipartite graphs) that differ along 3 dimensions (Table 1 and Fig. 5): *semantic mapping*, *granularity*, and *expressivity*. As we show for *Romeo and Juliet*, the representations we provide emphasize different aspects of the underlying raw data (Figs 2–4, 6), and they yield widely varying results even for simple measurements of character importance (Figs 7–10). Thus, HYPERBARD *enables* and *demonstrates the need for* research on how representation choices impact the outputs and performance of graph learning, graph mining, and network analysis methods.

C.3 The critique

The Community is designed as a microcosm of *our community*, including all levels of academic seniority as well as common supporting roles. The characters *inside* the Community exhibit cognitive, behavioral, and interaction patterns that frequently afflict people with corresponding roles in our community. The characters *outside* the Community appear as their antidotes, challenging the status quo and engaging in free-spirited scientific inquiry. As the play progresses, CREATURE gets caught up between both worlds, and we witness the force of community dynamics acting upon individuals that do not fit in. Examples of community phenomena featured in the play (there are many more): a struggling PhD student (CREATURE), abuse of power and difficulties of criticism in hierarchical organizations (PROFESSOR), administrative overload at the top of the pyramid (PROFESSOR and SENIOR RESEARCHER), cynical resignation, disillusionment, and complicitness (COLLEAGUE), publish or perish (DEADLINES), academia versus "freedom" (Community versus forest), mental health (CREATURE attempts *suicide*), uncomfortable viewpoints being shut down (HYPERBARD is *cremated*).

Appendix D: Play documentation

D.1 Inspirations

The play deliberately adopts and adapts ideas and text fragments from Shakespeare's works and other popular texts. With reference to the full version of the play [5], these are:

- Dramatis Personæ: Three deadlines ∼ three witches from Shakespeare's *Macbeth*
- Induction: Framing device used in Shakespeare's *The Taming of the Shrew*
- Act I, SCENE II, l. 32: A phrase famously *attributed* to Martin Luther
- Act II, SCENE I, l. 127: Allusion to a series of sketches from Monty Python's Flying Circus
- Act II, SCENE III, ll. 159–179: Jon's speech from Shakespeare's *As You Like It*
- Act II, SCENE IV, ll. 184–191: Faust's speech from Goethe's Faust I
- Act III, SCENE I, ll. 303–316: Ariel's Song from Shakespeare's *The Tempest*
- Act III, SCENE I, ll. 319–332: Hamlet's monologue from Shakespeare's *Hamlet*
- Act IV, SCENE III, l. 370: Juliet addressing Romeo in Shakespeare's *Romeo and Juliet*
- Act IV, SCENE III, ll. 401–402: Pieces from Jon's interactions in Shakespeare's *As You Like It*
- Act V, SCENE I, ll. 416–429: Shakespeare's *Full Many a Glorious Morning Have I Seen* (Sonnet 33)
- Act V, SCENE II, ll. 424–432: Macbeth's monologue from Shakespeare's *Macbeth*

D.2 Style

Our layout follows the Oxford Shakespeare from 1916 [17] (whose text sometimes differs from the Folger Shakespeare underlying our data [14], especially in the stage directions). We adopt the basic language patterns characteristic of Shakespeare's plays, using primarily blank verse, i.e. non-rhyming verse in iambic pentameter with feminine endings allowed, but also prose and rhyming verse. Our main character switches between blank verse and prose depending on their internal state. Longer passages of rhyming verse occur in song and sonnet adaptations (see Section D.1); shorter passages of rhyming verse are scattered throughout the play. We generally use Modern American English, sprinkled with brief interludes of Old British English.