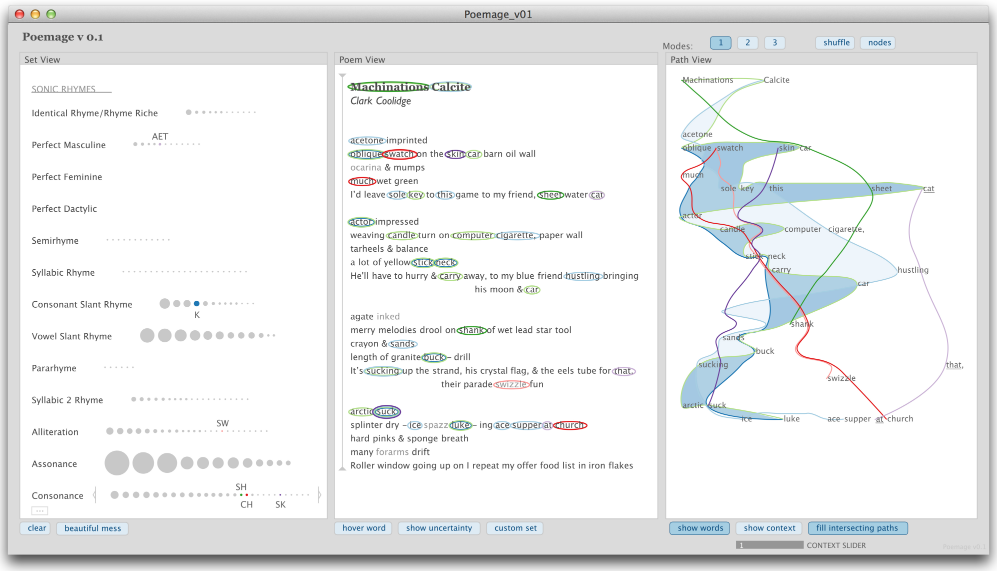
Poemage: Visualizing the Sonic Topology of a Poem

Nina McCurdy, Julie Lein, Katharine Coles, Miriah Meyer



Machinations Calcite
Clark Coolidge

acetone imprinted
oblique swatch on the skin car barn oil wall
ocarina & mumps
much wet green
I'd leave sole key to this game to my friend, sheet water cat

actor impressed
weaving candle turn on computer cigarette, paper wall
tarheels & balance
a lot of yellow stick neck
He'll have to hurry & carry away, to my blue friend hustling bringing
his moon & car

agate inked
merry melodies drool on shank of wet lead star tool
crayon & sands
length of granite buck – drill
It's sucking up the strand, his crystal flag, & the eels tube for that.
their parade swizzle fun

arctic suck
splinter dry – ice spazz duke – ing ace supper at church
hard pinks & sponge breath
many forarms drift
Roller window going up on I repeat my offer food list in iron flakes

Fig. 1. The Poemage interface comprises three linked views: *(left)* the set view allows users to browse sets of words linked through sonic and linguistic resemblances; *(middle)* the poem view allows users to explore sonically linked words directly via the text; *(right)* the path view shows the sonic topology of a poem.

Abstract—The digital humanities have experienced tremendous growth within the last decade, mostly in the context of developing computational tools that support what is called *distant reading* — collecting and analyzing huge amounts of textual data for synoptic evaluation. On the other end of the spectrum is a practice at the heart of the traditional humanities, *close reading* — the careful, in-depth analysis of a single text in order to extract, engage, and even generate as much productive meaning as possible. The true value of computation to close reading is still very much an open question. During a two-year design study, we explored this question with several poetry scholars, focusing on an investigation of sound and linguistic devices in poetry. The contributions of our design study include a problem characterization and data abstraction of the use of sound in poetry as well as Poemage, a visualization tool for interactively exploring the sonic topology of a poem. The design of Poemage is grounded in the evaluation of a series of technology probes we deployed to our poetry collaborators, and we validate the final design with several case studies that illustrate the disruptive impact technology can have on poetry scholarship. Finally, we also contribute a reflection on the challenges we faced conducting visualization research in literary studies.

Index Terms—Visualization in the humanities, design studies, text and document data, graph/network data

1 Introduction

The use of digital tools across disciplines in the humanities has exploded during the last decade. Popular projects such as the Google Ngram Viewer [37] and Wordle [55] have harnessed the power of computation to look across huge corpora of texts, leading to insights that had never been available before. Tools such as these are highly effective in supporting what is called *distant reading* — a term coined by literary scholar Franco Moretti to describe critical approaches that seek to understand literature and literary history by aggregating and quantitatively analyzing large text corpora.

Despite this new mode of scholarship, traditional humanities scholars continue to engage primarily in a very different type of analysis called *close reading*. As its name implies, close reading involves a detailed analysis of a text in all its complexity, encompassing an analysis not only of specific operations such as syntax, rhyme, and meter; such figures as metaphor and allusion; and such linguistic effects as affect, but also of how these operations interact across the temporal and spatial field of the text, with each other and with the reader, to create meanings greater than the sum of the parts. As this description suggests, much of the work done in close reading is well beyond the current capabilities of computation. Thus, the true value of computation to close reading is still very much in question and is the topic of an ongoing dialogue in the digital humanities. While a handful of computational tools have been designed to support close reading, much of the problem space remains unexplored.

We conducted a two-year design study with poetry scholars and practitioners to explore this gap. Our two primary collaborators, both of whom are co-authors on this paper, identify both as poets and as academics. We also engaged a network of practitioners, including two professors and two students of poetry. Together, these collaborators have literary expertise in medieval, early modern, modernist, and contemporary poetry, and they analyze poetry from a range of traditions and periods. Furthermore, they write formal verse, free verse, and experimental poems, and thus bring a diversity of theoretical viewpoints to their critical and creative work.

During this design study, we encountered several specific challenges that affected our design process. First, supporting close reading of poetry is a truly wicked problem [6, 49]: not only was it initially unclear as to *what* to visualize in a poem, but the design space for creating visual representations of poems and their features was completely open, since the use of technological tools as direct interventions in close reading (as opposed to in pedagogy and instruction) is still almost unknown to literary scholars. The second challenge we faced was that our collaborators belong to a community that sees the integration of technology into their research practices as largely unnecessary and potentially even intrusive. These challenges motivated us to use a highly collaborative and exploratory design process that takes the same experimental and even playful approach that our collaborators exhibit when reading and writing poetry.

Our design approach not only enabled us to learn more about poetry and close reading, but also disrupted our collaborators' view of poetry, pushing them to develop new perspectives on how poetic devices within a poem work together to create a response in the reader. These new perspectives led us to consider the *topology* of a poem, the complex structures formed from the interaction of sets of words across individual poems. Specifically, within a given poem we consider sets of words with similar sonic patterns. We focused our visualization design efforts on capturing poetic topology and providing a canvas for poets to explore the complex structure of sonic devices within a poem.

The specific contributions of this work are a characterization and abstraction for visualizing sonic devices in poetry; an open-source implementation of a tool for visualizing the sonic topology of a poem, called Poemage and shown in Figure 1; validation of this design study through several case studies that illustrate the efficacy of Poemage for not only providing novel analysis insights but also enabling the creation of new poems and literary ideas; and a reflection on the unique nature of conducting visualization research in literary studies.

2 Previous Work

A number of highly effective tools exist in support of distant reading. Synoptic text visualization tools like GistIcons [18], Docuburst [13], Compus [25], and Galaxies [58] employ semantic analysis to extract key concepts and allow users to gain quick overviews of one or more documents and to run comparisons across large bodies of text. Tag cloud-based tools like Wordle [55], TextArc [45] and the variant Parallel Tag Clouds [15] provide a different kind of summary by focusing on the frequency and distribution of individual words or phrases. Several more sophisticated tools [52] [16] provide broad overviews while also allowing users to explore finer level connections. In general, these tools treat a given text, or texts, as a bag of words, on which they perform a range of analysis without regard to structural and semantic context, features that are critical to the interpretation of poetic text.

Diverging from this slightly, FeatureLens [20] includes the repetition of expressions, revealing interesting patterns within and between documents. Techniques such as Phrase nets [54], Arc Diagrams [56], and The Word Tree [57] present more complex patterns based on a range of relationships. Building on The Word Tree, WordSeer [43] facilitates exploratory analysis of literary text.

Several visualization tools exist for specifically analyzing poetry. PoemViewer [2] employs rule-based visual mapping techniques to present a range of information about the poem, from traditional rhyme patterns to low-level sentiment analysis. PoemViewer attends to sound on a much deeper level than many tools, not only visualizing various types of phonetic repetition such as end rhyme, internal rhyme, assonance, consonance, and alliteration but also providing information about the physiology of sound production. While PoemViewer provides a wealth of information, both structural and relational, its interface does not capture the dynamism of a poem to the degree that our collaborators would like; poetic elements are, for the most part, presented as isolated objects, and poems are portrayed as static systems.

A second visualization tool, Myopia [8], was designed to aid in the close readings of poems. Myopia attends to a broad range of poetic elements, from meter, sound, and syntax to metaphor, personification, and emotion. These elements, however, must be coded a priori, a task that is currently done manually by a poetry expert and thus limits the tool's usefulness to a handful of poems. Although our analysis is limited to sound, Poemage processes text automatically, allowing users to explore any poem of their choosing, and for a broader range of sonic patterns.

The sonic analysis aspect of our research is closely related to Tanya Clement's seminal work on the analysis of aural patterns in text [10] and the exploration of the distance between the eye and the ear [9]. The visualization tool ProseVis [9] allows users to interactively explore the sonic transcription of a text and aids in the discovery of sonic patterns on different levels of granularity. Whereas ProsVis, and to a slightly lesser degree, Myopia and PoemViewer, capture and visualize the individual components of sound, Poemage extracts a range of more complex sonic patterns from a given poem and visualizes the interaction of such patterns across the space of the poem.

3 Design Process

Our primary collaborators had participated in previous visualization research [2], which acted as a first step towards overcoming their resistance to integrating technology into their own practices. Their resistance was rooted in part in an anxiety that the computer would inhibit the qualitative experience of the poetic encounter and in part a skepticism that it would be possible to visualize the interaction of any set of poetic features at a level of complexity that would allow them to make new and interesting observations. This initial project, however, left them deeply intrigued. While there was some remaining resistance, they approached this design study with *"skeptical enthusiasm"* to see if it is possible to use technology to probe more deeply, beginning with a high-level investigation into sound, into questions of what makes a poem a poem and how a poem does what it does. Their larger goal is to create a tool that will be of use to the broader poetry community.

Integrating computation into the practice of close reading is a wicked problem [6, 49]. To quote a prominent critic of the digital humanities, Stanley Fish, *"You don't know what you're looking for and why you're looking for it, how then do you proceed?"* [26] Thus, we initially spent significant up-front time in joint conversations with our two primary poetry collaborators to determine what their goals were and how they imagined a visualization tool affecting their experience of a close reading. These sessions took place, at a minimum, on a monthly basis, were held at the University of Utah, typically lasted from two to three hours, and were often recorded for future reference. One primary collaborator, who was non-local for the majority of this collaboration, participated remotely via video conferencing.

The initial conversations were broad and open-ended: the poets did not have specific goals, they did not want a tool to *"solve"* a poem [48], and they described a wide array of poetic devices, such as affect, imagery, sound, pun, and metaphor, that they look at in a close reading. Our collaborators presented examples of interesting features and interactions within poems they had previously studied. In parallel, we investigated established methods for computationally detecting and analyzing the devices that most interested them. For many of these devices, the level of analysis that was of particular interest to our collaborators was beyond current technological capabilities, such as the detection of metaphor and imagery. The exception to this was sound, which is detectable, with some limitations, by established computational linguistics techniques.

Once we, as a group, decided to focus on sound, our attention turned to developing a system that would automatically sonify a poem. Building on existing approaches for the sonification of text, we developed a formalism for analyzing sonic devices in English-language poetry [40]. The formalism describes sonic patterns as rhyme, with the definition of rhyme being one that is both broad and flexible — rhyme is a poetic device that varies in definition from poet to poet. Our formalism includes a language for expressing a broad range of visual and sonic rhyme types and an associated ASCII notation designed for poets. In addition, we developed an open-source implementation of our formalism, a tool called RhymeDesign, initially as a platform to test and improve our formalism and eventually as a tool for poets to explore custom sonic devices in poetry. This software subsequently supplies the back-end to our visualization tool Poemage.

Even after determining what sonic data we wanted to explore, considerable design challenges and open questions remained. Close reading covers a broad range of tasks, encompasses varying styles of analysis, allows many different points of entry, and accepts an extensive range of sometimes radically divergent interpretations. In addition, our collaborators admitted resistance to integrating technology into their close reading. Thus, we also had to cultivate their trust, commitment, and enthusiasm.

A highly collaborative and exploratory design process proved to be critical in helping us navigate these challenges. We began by discussing the poets' experience with, and the results of, their previous visualization research. Next, we employed a number of different techniques in an attempt to clarify our point of entry. The first technique was an observation of a pair of close readings between our two primary collaborators, starting with the poem "Prayer" by Jorie Graham, followed by a close reading of "Night" by Louise Bogan. Close readings can be performed internally by one poet or externally as a conversation between two or more people. Throughout many of our future conversations, our collaborators returned to "Night" and other poems and picked up close readings in order to illustrate particular concepts — such as how sonic patterns can reinforce or undercut semantic meaning. Other techniques for clarifying our entry point included studying an annotated poem from one of our collaborators, giving our collaborators a list of potentially interesting sonic devices that could be detected computationally and having them compile a list articulating the various sonic features that they were interested in exploring, and attending public poetry readings to better understand the nature and practices of the poets and poetry scholars.

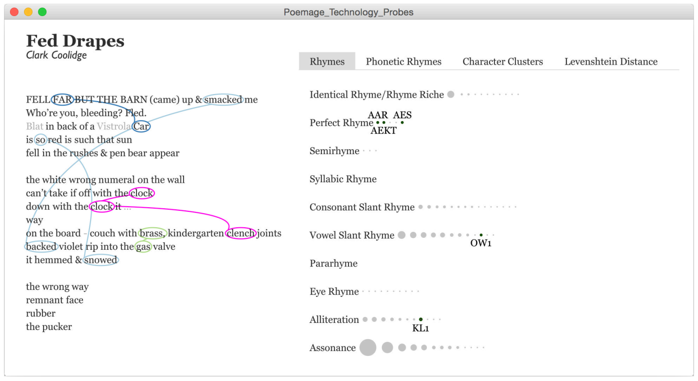
Based on these activities, we ideated on a range of design possibilities to pursue, which we then developed into a set of technology probes [32] — we discuss details of these probes in Section 4. The probes were successful both in engaging our collaborators and also in helping us better understand the problem space. We iteratively refined the probes over the course of several months based on extensive user feedback, both casual and via formal interviews, from our primary collaborators as well as our extended network of poets and poetry scholars. The incremental steps and the adjustments we made in response to their feedback and critiques helped the poets become familiar with the technology and also resulted in an interface that reflected their interests, aesthetics, and values. In addition, because our meetings were highly conversational and interactive, the poets actually generated poetic insights in our meetings on the fly, simply in response to developing and imagining the tool. This gave them confidence that the work, and eventually the visualization tool, would be useful to them.

Results from the technology probes formed our initial design ideas for the tool Poemage. These ideas were implemented into an initial prototype and presented to our primary collaborators. Based on casual feedback, we refined and improved existing features and added new features, the details of which are provided in Section 7.

4 Technology Probes

The technology probes were implemented in Processing [27] and combined into a single, multi-tabbed interface, shown in Figure 2. Users would load a poem of their choosing into the interface, which displayed the text of the poem, along with information about selected sets of sonic patterns. Following an initial development period in which versions of the probes were presented for informal feedback to our primary collaborators, the technology probes were deployed, along with written documentation, to four of our collaborators. The collaborators were given approximately one month to experiment with the probes, after which formal interviews were conducted. Interviews included brief observations of our collaborators using the tool, followed by questions surrounding approach, capabilities, and general usability. Interviews were recorded and transcribed, and one observation period was screen captured.

The initial goal of the technology probes was to explore the many different aspects of sound within a poem, as well as the role that sonic analysis plays in close reading. Using these probes, we experimented broadly in order to better understand, and to help our collaborators better understand, what kinds of sonic relationship they were interested


Poemage_Technology_Probes

Fed Drapes
Clark Coolidge

FELL FAR ~~BUT THE~~ BARN (came) up & smacked me
Who're you, bleeding? Fled.
Blat in back of a Vistrola Car
is so red is such that sun
fell in the rushes & pen bear appear

the white wrong numeral on the wall
can't take if off with the clock
down with the clock it ...
way
on the board - couch with brass, kindergarten clench joints
backed violet rip into the gas valve
it hemmed & snowed

the wrong way
remnant face
rubber
the pucker

Fig. 2. Interface for the first set of technology probes.

in exploring in a poem. What we found led us to develop a broader understanding of rhyme, which we discuss further in Section 5. Furthermore, these initial probes indicated to us that our collaborators were not interested in exploring individual sonic relationships, but instead they sought to understand how different sonic patterns interact and evolve across a poem. We thus developed a second set of technology probes to explore this notion of sonic topology. These investigations were instrumental to the development of our data abstraction, presented in Section 6.

The technology probes also allowed us to establish a common vocabulary with our collaborators; to focus on understanding how to capture data from a poem, as opposed to how to visualize it; and to define the space of what we could computationally detect in a poem. Overall, the probes helped us create an experimental and playful research environment that we maintained for the duration of the collaboration.

5 Poems and Sound

Poets and scholars see poems as living and relational, their literary features interacting not only with each other but also with us as readers. In close reading, a poetry scholar carefully attends directly to specific texts, tracing the interactions among such literary features as rhyme and meter, sound, figures, and syntax, while also considering how a given poem explicitly or implicitly converses with other poems in the literary canon. Although not viewed as an established technique for writing poetry, the experience of close reading often leads to the generation of new poems, and many poets do engage it as a prod to composition.

As a broad, literary device, sound provides poets with a rich source of play and can deeply influence the interpretation of the poem. Because of its emotional power and the way it works directly on the body of the reader of the poem, sound is an important source of poetic potency and can be used to reinforce or to undercut meaning conveyed via other poetic devices. In addition, sonic ambiguities — for example, in homographs like *wind* and *bow* as well as in words with multiple pronunciations — also help generate multiple possible interpretations of the same poem. Furthermore, unlike many devices which may or may not be present at a particular moment or even at all in a given poem, sound is arguably pervasive in every poem at all levels.

Our collaborators consider a broad range of sonic and sound-related devices in their close readings of poems: from traditional types of rhyme such as *rhyme/sublime* and *picky/tricky*; to patterns involving the spellings of words, including eye rhymes (*cough/bough*) and anagrams (*desserts/stressed*), which may or may not relate sonically; to patterns surrounding the physiological production of speech sounds, such as the location of the tongue in relation to the lips. In this design study, we refer to all sonic and linguistic devices as *rhyme* [40], a broad definition embraced by our collaborators.

Our collaborators are particularly interested in the conceptual metaphor of a poem as a flow [36]. By approaching a poem, for the purposes of visualization, as a fluid moving via its linguistic devices and figures through a defined space, the flow metaphor captures three distinct levels of poetry analysis: the movement of individual linguistic and sonic devices through the space of the poem, how the interaction among such devices contributes to the complex sonic-temporal structure of the poem, and the impact that individual flows and collections of flows have on their surrounding region. In addition, flow introduces the notion of sonic turbulence — a metaphor for locations in a poem where there is increased intensity, energy, and activity due to the interaction of poetic devices. Over the course of our collaboration, we worked to translate this conceptual metaphor first into a data metaphor, expressed in terms of the extracted sonic patterns, and then into a visual metaphor, visually encoding the features and characteristics captured in our data metaphor. We discuss our data metaphor in Section 6 and our visual metaphor in Section 7.

6 Abstraction

In order to visualize the flow of a poem, we translate this metaphor into three data components: *poemspace*, *rhyme sets*, and *sonic topology*. The first component, **poemspace**, is the 2D space of the poem as it is printed on a page. Our collaborators were adamant about the importance of maintaining the spatial and textual context of the poem itself for two reasons. First, a poet can play with whitespace and layout to encode or enforce some sort of meaning in the poem. Second, any sort of data we pull out from a poem computationally will (usually) be meant to augment the reading of the poem itself. Poemspace, however, is unique compared to other 2D spaces as the reading of the poem constrains the way that movement within poemspace can happen — left to right, followed by top to bottom. In poemspace, each word has a location based on where it falls within a line, and where that line falls within a poem.

Words are related to each other not just spatially within poemspace, but also based upon their similarities to other words with respect to some sonic pattern. For example, in Figure 1, the words *cat*, *that*, and *at* (underlined) are in a set together because they form a user-selected perfect rhyme. A set of words linked by a rhyming scheme is called a **rhyme set**. In this set, the words are ordered based upon their location in poemspace. Each word in a poem can belong to none, one, or many different rhyme sets, depending on which rhyme schemes are defined.

The **sonic topology** of a poem is represented by the distribution of rhyme sets across a poem and how those sets of words interact with each other, or not. From the conceptual metaphor of a poem as a flow, the places of sonic turbulence in a poem exist where multiple rhyme sets intersect, i.e., a word exists in multiple sets. To capture the sonic topology, we create paths from the rhyme sets, where each word in a rhyme set is connected by a link based upon their order in poemspace. For example, in Figure 1, the set including *machinations*, *calcite*, and *oblique* is ordered from top to bottom, left to right.

Our collaborators noted several different types of path interactions that are of interest, each of which is illustrated in Figure 3:

- *intersecting*: paths intersect at a single node.
- *overlapping*: paths intersect at multiple consecutive nodes.
- *merging*: paths intersect and then overlap.
- *diverging*: paths overlap and then split.
- *emerging*: paths begin at a point of intersection.

At a low-level, our collaborators are interested in identifying and exploring places of turbulence, indicated as intersecting, merging, diverging, and emerging paths. At a higher level, they want to understand the places of turbulence within the context of the poem, and in the context of other poetic devices they identify in the course of their close reading.

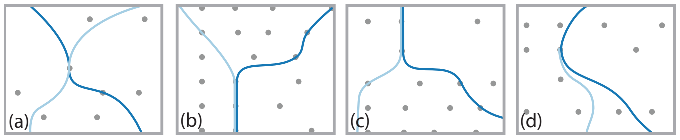


Fig. 3. When rhyming sets are represented as paths, several interesting interactions can occur: *(a)* intersecting paths, *(b)* merging paths (also displaying overlap), *(c)* diverging paths (also displaying overlap), *(d)* and emerging paths.

As described in Section 5, sonic ambiguity exists for words with multiple pronunciations. The implementation of our formalism for describing rhyme captures this ambiguity and stores multiple versions of rhyme sets based on alternate pronunciations. For our collaborators, this ambiguity is a source of great joy as it represents possible alternate or additional meanings, and so enriches the interpretive experience of reading the poem. As such, we can describe the set of paths that results from sampling the ambiguous pronunciations as an ensemble of possible paths through a poem. Our collaborators are interested in exploring this ensemble to probe for different sonic topologies and different poetic interpretations.

7 Poemage

Our second contribution is the design and implementation of Poemage, a visualization tool for interactively exploring the sonic topology of a poem. The Poemage interface, shown in Figure 1, comprises three linked views: the set view *(left)*, the poem view *(middle)*, and the path view *(right)*. Multiple views provide users with multiple entrances into the poem, a feature expressed repeatedly by our collaborators in the technology probes as being highly effective for gaining new perspectives and insights. In addition, multiple views allow users to manipulate and play with the text, an example of which is presented in Section 8.2, and to view abstracted representations of the poem while maintaining a close connection with its original form.

A user's session with Poemage begins with the selection of a poem of interest, which is loaded into the tool via a text file. Poemage preprocesses the poem and creates the rhyme sets based on 24 different rhyme types built into the tool. For the current version of the tool, we worked with our collaborators via the technology probes to define a set of rhyme types that captures the majority of interesting sonic patterns in a poem — Table 1 lists the types of rhymes currently supported within Poemage. We note that the back-end of Poemage includes a formalism for defining rhyme that can be modified through the open-source release of the software, thus supporting a definition of rhyme much broader than those available in other poetry visualization tools.

In this section, we describe the design and capabilities of the individual views and of the interface as a whole. As part of the path view description, we present two novel extensions of existing graph visualization techniques. Poemage was implemented in Processing [27], and the source code is freely available at `http://poemage.org`.

7.1 Set View

The set view allows users to browse through the various detected rhyme sets for a given poem. Each circle represents an individual rhyme set, the radius of which encodes the relative number of words participating in a given rhyme set. Sets are organized by rhyme type and are ordered by decreasing set size to make the largest sets most visible. The rhyme types are organized into *sonic rhymes*, which involve matching patterns in sound, and *visual rhymes*, which involve matching patterns of alphabetic characters. Collapse and expand buttons located to the left of each category header allow users to omit either of these categories from their exploration. We observed the necessity to explore visual and sonic rhymes together, and separately, in the use of our technology probes.

The set view supports browsing of specific rhyming sets. When a user hovers over or selects an individual set or entire rhyme type, a pop-up label indicates the specific rhyme pattern for the set, and the associated rhyme sets are also displayed in the poem and path views. Color is used to link sets and rhyme types across views. For our color scheme, we rotate through an adapted version of giCentre's *12-class categorical paired* colormap [1]. The notion of browsing was something that our collaborators responded very well to in the technology probes. Not only did it come naturally to them, but it also provided a slightly abstracted way of exploring the poem while maintaining a close connection to its original form.

Clicking the *beautiful mess* button at the bottom of the set view selects all rhyme sets. This feature was first requested by our collaborators in one of the technology probes, and subsequently became

Table 1. Rhyme types implemented in Poemage for creating sets of sonically related words.

Rhyme type	Description	Example
Identical rhyme	match in all sounds. Includes repeated words and homographs	pair/pair; pare/pair
Perfect rhyme:	matching stressed vowels sound and all proceeding sounds	
Perfect masculine rhyme	stress on the final syllable	rhyme/sublime
Perfect feminine rhyme	stress on the second to last syllable	picky/tricky
Perfect dactylic rhyme	stress on the third to last syllable	gravity/depravity
Semirhyme	perfect rhyme with additional syllable on one word	end/defending
Syllabic rhyme	perfect rhyme between stressed and unstressed syllables	wing/caring
Consonant slant rhyme	matching trailing consonants of stressed syllables	and/bent
Vowel slant rhyme	matching vowel sounds of stressed syllables	eyes/light
Pararhyme	matching leading and trailing consonants of stressed syllables	tell/tail/tall
Syllabic 2 rhyme	rhyme between initial stressed syllables	restless/westward
Alliteration	matching leading consonant sounds of stressed syllables	languid/lazy/line/along
Assonance	matching vowel sound (independent of stress)	blue/estuaries
Consonance	matching leading and/or trailing consonant sound (independent of stress)	shell/chiffon
Forced rhyme	perfect rhyme with imperfect match in final consonant sounds	shot/top/sock
Eye rhyme	spelling indicates perfect rhyme but sounds do not match	cough/bough
Character clusters	matching substring involving 1-4 characters	restless/westward
Mixed character clusters	mixed substring involving 2-4 characters	inlets/itself
Anagram	words formed out of the same set of characters	nights/things
Phonetic alliteration	leading consonants of stressed syllable match in mouth placement	pen/boy
Phonetic assonance	vowels of stressed syllables match in mouth placement	edible/anchor

one of their favorite features as well as a surprisingly valuable addition to the tool. Despite our initial hesitation to support showing all rhyme sets at once due to visual clutter, our collaborators were able to make interesting discoveries with this feature. One such discovery was the single, isolated pronoun *you* in the poem *"This Is Just to Say"* by William Carlos Williams. The beautiful mess revealed that *you* was the only sonically unconnected word in the poem, as shown in Figure 4. This insight was particularly powerful to our collaborators given the poem's occasion: to see *you*, the addressee and recipient of the ostensive poetic apology, excluded from the poem's many sonic relationships sharply heightened their sense of the poem's ambiguities. This example demonstrates how sound can work with and against semantics to elaborate readers' potential interpretations, and even affective experiences, of poems.

7.2 Poem View

We designed the poem view to support direct exploration of poetic devices in a poem's original form. Similar to hovering and selecting rhyme sets in the set view, users can hover over and select words in the poem view, which in turn selects all the rhyme sets for which the word is a member. Browsing through words in the poem was requested by our collaborators early on and proved to be a very natural and effective way for them to interact with the text.

When rhyme sets are selected either via word selection in the poem view, or directly in the set view, ellipses are drawn around the words in the selected set. The color of each ellipse corresponds to the assigned set color, as described in the previous section. Although we explored several different ways to encode selection, our collaborators preferred the ellipse, as it reminded them of their own annotation practices. For words belonging to multiple selected sets, concentric ellipses form a bullseye, similar to the concentric rings employed in LineSets [3], and provides a quick overview of the set membership for a given word. One collaborator commented that this encoding appeared to her as pebbles being dropped in a pond, with heavier pebbles causing more ripples, a nod towards a visual metaphor of the flow of a poem.

Clicking the *custom set* button allows users to build custom rhyme sets by selecting specific words in the poem view. Similarly, clicking the *show ambiguity* button highlights words with multiple pronunciations and allows users to select alternative pronunciations. Scrolling is enabled for poems of longer length with a print-to-pdf keyboard option providing a complete view of the poem and visualization.

7.3 Path View

The sonic topology of a poem is visualized in the path view, an abstract view that represents words in a poem as nodes at their corresponding location in poemspace. Our decision to map words to nodes, rather than to smaller linguistic units such as syllables, was rooted in observations of our collaborators during the technology probes and in the observed close readings consistently tracing sonic devices back to the semantics of the involved words. When a user selects a rhyme set in either the set or poem view, the associated path connecting the words in the rhyme set is shown in the path view as a curve connecting the associated nodes. We explored a variety of ways to represent the paths as node-link diagrams and found this representation best captured the characteristics of our flow metaphor.

We provide context in this abstracted view through several mechanisms. First, the path view is linked with the set and poem views such that selection and highlighting in the other views causes paths to appear in this view. Second, when a user hovers over a node in the path view the corresponding word appears as a pop-up. Third, clicking the *show words* button displays the set words associated with a given path. Fourth, clicking the *show context button* displays nearby words surrounding a given path. The extent of surrounding words can be adjusted via the *context slider*. An example visualization employing these features is shown in Figure 5 *(c)*.

Rendering paths in poemspace requires a number of design considerations: rerouting paths to avoid ambiguous set membership of words, effectively rendering multiple paths at once, supporting multiple interpretations of poemspace, and incorporating sonic ambiguity. We discuss each of these considerations in more detail below.

We developed a number of prototypes to explore design variations for this view, starting with images we shared with our collaborators generated from off-the-shelf node-link diagram tools [23] [4]. Based on feedback about these tools, we designed a path visualization that resembles other line-based overlay techniques such as LineSets [3], Kelp diagrams [19], and KelpFusion [42], in which the spatial context of the set data is preserved and shortest path algorithms are employed to determine routes linking set members. Like KelpFusion, our approach combines line-based with region-based overlay techniques [14] [7] [30]. Rather than use convex hulls to delineate sets, however, our path visualization employs fill to emphasize regions enclosed by sets intersecting at multiple nodes.

7.3.1 Rerouting Paths

In our prototypes, we encountered problems with edges intersecting words that were not included in the path. Furthermore, examples like

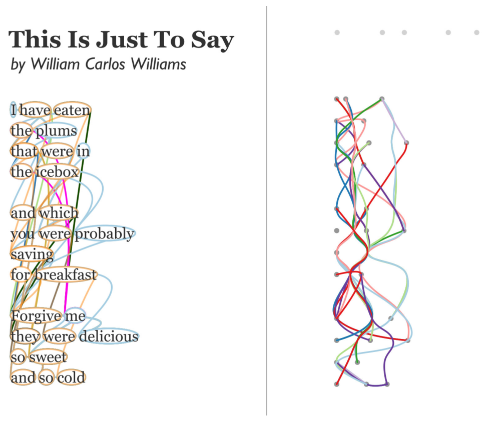


Fig. 4. The *beautiful mess* highlights the sonic isolation of the word *"you"* in this poem. A visualization from a technology probe obscures this isolation *(left)*, whereas rerouting in Poemage reveals the anomaly *(right)*.

the *you* anomaly that we discussed in Section 7.1 highlighted the importance of having isolated nodes appear isolated. As such, we decided to *reroute* edges such that they explicitly avoid words not included in a path.

We experimented with several different rerouting [39] and bundling [17] [24] techniques before finalizing our design. In the process, we discovered that we could take advantage of the vertical, regular spacing of poemspace to establish a simple and general rerouting technique. Our technique reroutes edges connecting words that are separated by more than one line in the poem, as these are the edges that may intersect words *not* in the path's set, as shown in Figure 5 *(a)*. For these edges, at each line of the poem that the edge intersects, we determine the closest *whitespace* to the edge intersection, i.e., the closest space between words. We place a new control point at the center of these whitespaces and render the edge as an interpolating cubic bezier curve. This rerouting produces a meandering curve that avoids all words that are not included in the path, illustrated in Figure 5 *(b)* and *(c)*. A side-effect of this rerouting technique is that similar paths are naturally bundled together.

This rerouting technique integrates several different aspects of the conceptual metaphor of a poem as a flow: the notion that adjacent flows tend to aggregate, intensifying the same path, as well as the idea that flows can behave like eddies, bending and diverging dominant courses, disrupting their surroundings, looping backwards, dissipating, or developing in new directions. In addition, the rerouting generates much more organic, aesthetic curves than those generated in our previous implementations, which increased the overall efficacy of the tool for our collaborators. While we appreciate that minimizing wiggles is a common constraint in graph drawing, we claim that our approach improves, rather than obscures, our visual representation of poemspace. Our rerouting technique is most similar to techniques that use grid-based rerouting to bundle edges and reveal high-level patterns [35] . The inherent grid-like qualities of our data make our rerouting technique a more intuitive approach and allows us to avoid issues associated with standard grid-based methods.

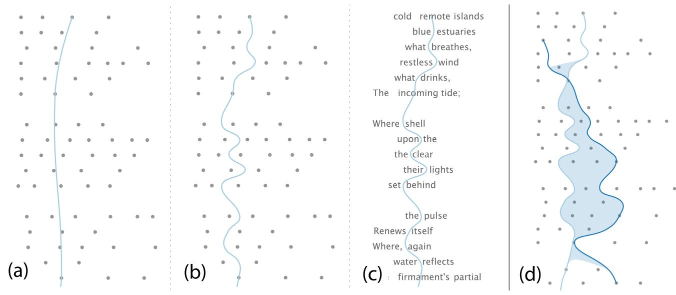


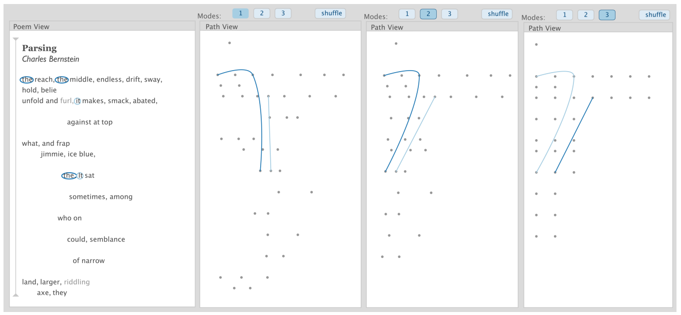
Fig. 5. Rendering the rhyme set paths in poemspace can lead to ambiguous set membership *(a)*. Poemage incorporates a path rerouting technique to disambiguate words *(b)*, as well as context information for the nodes *(c)*. *(d)* Fill function between two intersecting paths.

7.3.2 Drawing Multiple Paths

As we discussed in Section 6, of particular interest to our collaborators was exploring places in the poem where paths overlap, merge, diverge, and emerge. Our path rendering algorithm supports visualizing these locations through two mechanisms: maintaining a consistent ordering of paths to help users trace paths across a poem and a fill-function that emphasizes mergence, divergence, and emergence.

The ordering task closely resembles the metro-line minimization problem [46] with a unique combination of constraints — rerouting, bezier splines, and intermediate nodes. We adapt an existing technique to address these differences [44] by first calculating ordering for pairs of paths sharing common subpaths and then iteratively adding one path at a time such that the pairwise orders are maintained. Path order is computed on-the-fly as users select and deselect paths of interest. We disable ordering for the *beautiful mess*, as it causes significant visual clutter and obscures isolated nodes left exposed via rerouting.

In addition to interactive path ordering, we include a semi-transparent fill function to emphasize path mergence, divergence, and emergence, as shown in Figure 5 *(d)*. In places where paths intersect at multiple nodes, the fill spans the area enclosed by the two paths, and in places where two paths merge, diverge, or at single points of intersection, the fill approaches the intersection point. This fill technique is inspired by our collaborators' belief that interacting flows influence their surrounding region (and vice versa), sometimes pulling surrounding words closer together, other times pushing them apart — the fill seeks to reveal possible regions of influence formed via the intersection of paths. Fill is computed interactively on a pairwise basis. The user has the option of setting the fill to a constant color or to a color that is dependent on the involved paths.



Parsing
Charles Bernstein

the reach, the middle, endless, drift, sway,
hold, belie
unfold and furl, it makes, smack, abated,

against at top

what, and frap
jimmie, ice blue,

the, It sat

sometimes, among

who on

could, semblance

of narrow

land, larger, riddling
axe, they

Fig. 6. The three modes of poemspace shown in the context of the poem "Parsing" by Charles Bernstein: *(left)* original form, *(middle)* compressed whitespace, *(right)* and even spacing.

7.3.3 Deforming Poemspace

Another design concept that we experimented with was that of added whitespace. Inspired by poets like E.E. Cummings and Charles Bernstein who use whitespace to augment the shape of their poetry, our collaborators found that comparing the shapes of paths with and without added whitespace helped form interpretations as to why an author may have formatted the text in the way he or she did. In the path view, we support three different deformations of poemspace: the original form, compressed whitespace to just a single character width, and evenly spaced nodes. We illustrate these deformations in Figure 6. Buttons along the top of the path view allow users to toggle between these deformations.

7.3.4 Ambiguity

The final feature in the path view addresses the concept of ambiguity. In addition to allowing users to select alternative pronunciations for homographs and other multi-pronunciation words in the poem view, a *shuffle* button reruns the entire program based on randomly selected pronunciations, thereby sampling the ensemble members described in Section 6. This shuffle is meant to visualize and inspire different interpretations of the same poem.

8 Validation

We present two forms of validation for this design study. First, four case studies with our collaborators illustrate how Poemage not only supports novel analysis insights (Section 8.1), but also how the tool supports making and remaking new poems (Sections 8.2 and 8.3). Poemage was introduced to the poets via demos highlighting the various features and interactions, many of which were previously familiar to them from earlier prototypes. These demos were either given in person or recorded. Following the demos, the collaborators were given a week to experiment with the tool, after which interviews were conducted, recorded, and transcribed to gather user feedback. While semi-structured interviews were planned, in all three cases the opening sequence of questions *"Can you walk me through how you used the tool?"* and *"Did you gather any new insights, and if so, can you show me how you arrived at them?"* propelled a dialogue in which the remaining questions were answered and many additional topics were approached. Two of the poets kept journals of their experimentation [51], which provided direct narration for three of these case studies.

As a second form of validation, we discuss the impact that our collaboration has had on the scholarship of our direct collaborators (Section 8.4) — we argue that disrupting the thinking of these poets is an important mark of success for this work.

8.1 Close Reading with Poemage

One collaborator described her approach to using Poemage in analyzing a poem as *"noodling,"* hovering over one sonic feature after another in the set view and poem view, selecting and deselecting rhyme sets almost arbitrarily. She said her greatest successes and insights came in every case when she happened on something indirectly, through idle play — as she says, *"almost out of the corner of my eye."*

A specific example of this was an insight gained when glancing at the placement of nodes in the path view for the poem "Night" by Louise Bogan. While the placement of nodes in this poem is mostly regular in that there are generally a similar number per line (around 4) and they are mostly at similar distances from each other, indicating that there is typically about the same number of words per line and these words are of similar length, there was one line that had only two nodes, the second following very closely on the first. Thus, the abstracted view of poemspace revealed an immediately visible anomaly in the spatial distribution of words. This anomaly coincided with a powerful semantic moment in the poem, leading this collaborator to explore the rich sonic turbulence at that location and its connection and reinforcement of the semantic flow of the poem. She said that this view shed new light on a poem with which she was deeply familiar: *"In other words, not only is this the poem's turn, its pivot and crisis, but there's just a whole lot going on, a lot I wouldn't necessarily have considered in quite this way without the tool drawing my idle eye — a lot I hadn't in all these years considered up to this moment."*

She commented that Poemage took her into the poem in a different way than she was accustomed, and that this occurred via both the poem view and the path view. In a typical close reading, she begins with the title and first line and forges a slow, recursive path in which the overall movement is left to right and down the page, but in which a specific poetic event might send her back to the beginning of a line or up the page again as far as the title. An example of how Poemage changed this procedure for her occurs in the previously described observation of the node anomaly, which occurs late in the poem. Because of that observation, this specific encounter with the poem began with the first line of the final stanza, rather than at the beginning.

Another collaborator chose to explore Jorie Graham's poem "Reading Plato." She began by hovering through words in the poem view to see what overall patterns appeared in the path view. By scrolling up and down through the length of the poem, she was able to piece together a composite sense of the sizes and shapes of these patterns as they appeared and developed in the path view. Because she felt least likely on her own to discern specific examples of complex visual rhymes, she next turned her attention to that category of rhyme sets in the set view. She browsed through the mixed 4-character cluster rhyme sets, which immediately revealed some interesting results potentially relevant to her interpretation of the poem. A visualization of her selections is shown in Figure 7(c). She reflected on her exploration:

"The multiple-view interface felt engaging and responsive and it reflects the sensibility that I experience when reading a poem: that interpretive readings are made, choice by responsive choice, and that nothing is absolutely conclusive. Curves and complementary soft colors, mixed and blended through interaction, connote changeability and invite engagement without visually overpowering the user. The poem itself remains central both figuratively and literally while the multiple, flexible paths through the poem allow users to shift their focus quickly, between minute details and single patterns in isolation, or in relation to each other and the poem as a whole. In these ways, Poemage not only reveals patterns within the poem, but also enables users to see their own spontaneous choices, their own interpretive work and critical explorations in new ways — which in turn spur still further exploration."

8.2 Erasure Poetry

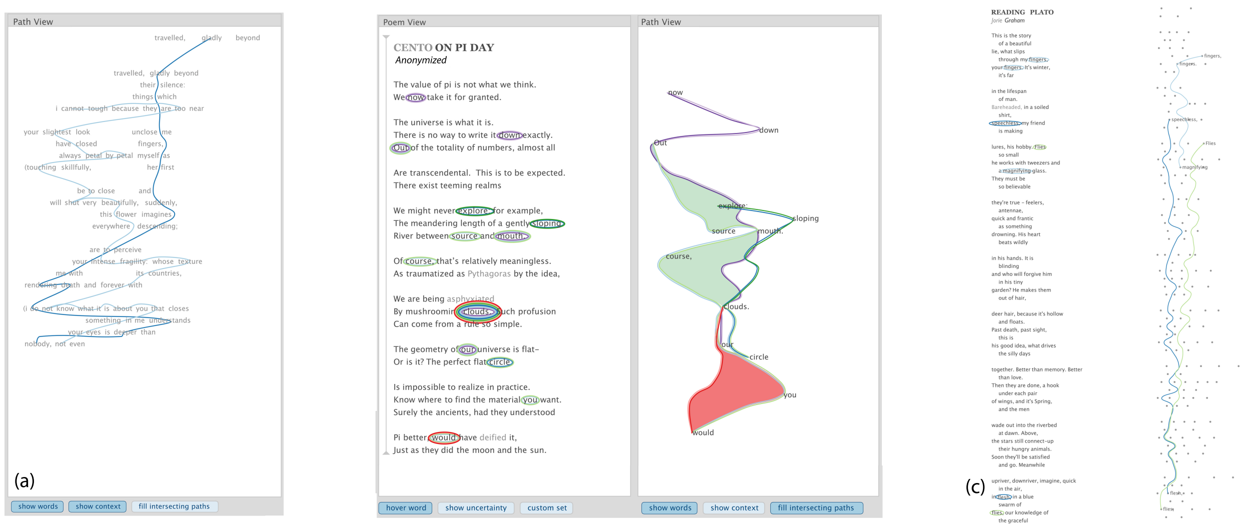
Erasure is a form of poetry generated by erasing words from an existing text, resulting in a new poem with potentially new meaning. The concept of an erasure was introduced by a collaborator from our extended network of poets in response to one of our technology probes. Iterating on the idea subsequently led to our inclusion of the *show words* and *show context* features in Poemage. These features allow users to explore all the possible erasures formed from single or combined rhyme sets and their surrounding regions in poemspace. This collaborator experimented quite a bit with the final implementation of these features in Poemage, and recently exhibited several of her erasure poems generated using the tool in a local art gallery, one of which is shown in in Figure 7(a). We describe her experience using Poemage to generate erasure poems here.

This collaborator took several different approaches to using Poemage. For poems she was deeply familiar with, her exploration was guided by previous observations and investigations, leading her to hover over and select particular rhyme sets in the set view. For less familiar poems, on the other hand, hovering over different words in the poem view, thereby revealing a word's various sonic connections with other words in the poem, helped her to quickly gain entrance to the text. Taking a slightly different approach, she also experimented with building interesting shapes in the path view by randomly selecting words and rhyme sets in the poem view and text view. Such selections were based on their visual impact on the visualization in the path view. This third approach allowed her to specifically investigate how introducing new sonic flows changed the sonic structure of the poem. In a similar fashion, she generated new erasure poems by enabling the *show words* feature and selecting one or multiple rhyme sets based both on the shape of their paths and the subpoems they revealed. She commented that the visualizations, and especially those using the fill function, went *"straight to the pleasure center of [her] brain."* She also commented that Poemage encouraged her to spend more time with the poem, which she felt was one of the biggest benefits of the tool.

8.3 A Cento in the (Re)making

A *cento* is a poem composed entirely of lines or passages taken from other authors. One collaborator used Poemage to explore a cento that she had composed by taking lines from an article in the New York Times written in honor of Pi day [53]. Her cento is shown in Figure 7(b). She loaded the cento into Poemage and proceeded, according to her usual practice, to noodle around, looking for various densities or lack of density. She also compared the cento to other poems entirely of her own making. What Poemage helped reveal to her was the extent to which she had managed to make another's text her own — how the poem *"looked"* sonically like one of hers. She recounts this:

"I noticed that my cento is in some ways as sonically intense as my poems built from scratch — with the notable difference that the sentences of a journalist tend to rely heavily on the verb 'to be,' a verb I use quite rarely when left to my own devices. 'Is' shows up in the cento as the dominant sound on exact rhyme, assonance, and consonance;



CENTO ON PI DAY
Anonymized

The value of pi is not what we think.
We now take it for granted.

The universe is what it is.
There is no way to write it down exactly.
Out of the totality of numbers, almost all

Are transcendental. This is to be expected.
There exist teeming realms

We might never explore: for example,
The meandering length of a gently sloping
River between source and mouth.

Of course, that's relatively meaningless.
As traumatized as Pythagoras by the idea,

We are being asphyxiated
By mushrooming clouds. Such profusion
Can come from a rule so simple.

The geometry of our universe is flat–
Or is it? The perfect flat circle

Is impossible to realize in practice.
Know where to find the material you want.
Surely the ancients, had they understood

Pi better, would have deified it,
Just as they did the moon and the sun.

READING PLATO
Jorie Graham

This is the story
of a beautiful
lie, what slips
through my fingers,
your fingers. It's winter,
it's far

in the lifespan
of man.
Bareheaded, in a soiled
shirt,
speechless, my friend
is making

lures, his hobby. Flies
so small
he works with tweezers and
a magnifying glass.
They must be
so believable

they're true – feelers,
antennae,
quick and frantic
as something
drowning. His heart
beats wildly

in his hands. It is
blinding
and who will forgive him
in his tiny
garden? He makes them
out of hair,

deer hair, because it's hollow
and floats.
Past death, past sight,
this is
his good idea, what drives
the silly days

together. Better than memory. Better
than love.
Then they are done, a hook
under each pair
of wings, and it's Spring,
and the men

wade out into the riverbed
at dawn. Above,
the stars still connect-up
their hungry animals.
Soon they'll be satisfied
and go. Meanwhile

upriver, downriver, imagine, quick
in the air,
in flesh in a blue
swarm of
flies, our knowledge of
the graceful

Fig. 7. User visualizations. (a) An example of an *erasure* of E.E. Cummings poem, *"somewhere I have never travelled, gladly beyond"* created by one of our collaborators and exhibited in a local art gallery. (b) A visualization showing a *cento* created by one of our collaborators. (c) A visualization of the poem *"Reading Plato"* by Jorie Graham, printed to pdf.

never in my own drafts. This initially made me a little despondent — should I give up making centos from the New York Times? — but another view of the poem shows me a wonderful set that included 'now,' 'down,' 'out,' 'source,' 'mouth' 'course,' 'clouds,' 'you,' and 'would,' and even encompassed 'circle,' which feels just like me. This raises a question: how is my own cento (I have several of them) like me and not like me? How do I unconsciously select sentences for the cento not only for their meaning but also for their sound? How does a cento in the making become more like me as I make it?"

This revelation, combined with other observations made using Poemage, inspired this collaborator to *"sonically reload"* her cento, rearranging the lines and passages based on the resulting visual representation in Poemage's path view.

8.4 A Disruptive Technology

One interesting, yet difficult to capture, measure of success for a design study is the act of disrupting the thinking of a domain expert [50]. Through the course of this design study, our poetry collaborators developed new perspectives on their domain and research practices through the lens of computation and visualization. Asking them to define precisely what is interesting in a poem, in a rhyme, and in a sound led to new thinking, which in turn enabled them to envision new ways of approaching a poem and to narrow the scope of their tasks from *"find everything interesting in a poem"* to *"visualize the sonic topology."*

One such insight was the re-articulation and development of the conceptual metaphor of a poem as a flow, into data and visual metaphors. As part of a session entitled "Things My Computer Taught Me About Poems," at the 2014 Modern Language Association's annual conference, our two primary collaborators presented on this metaphor, and specifically on the notion of turbulence and poetic time [12]. These collaborators also published significant articles in which they reflect on the impact that visualization research has had on their poetry scholarship. In one of these articles, the author focuses not on the technology itself but on how the need to teach the technology, and the computer scientists, what to look for and visualize in poems forced her to be much more precise in her own thinking about not only sonic devices and how they signify within poems, but also more complex questions of language and imagery [11]. The author of another of these articles describes how this collaborative research prompted her to re-read familiar poems in new ways long before our software was ready to explore, and discusses ways this research has led her to re-evaluate and re-imagine her theoretical positions on such literary questions as how poetic time operates and even what reading entails [36].

We put forth these results as an important validation of the impact this work has had on our collaborators' poetry scholarship. Since these interviews, we have continued to work with these poets to explore new extensions of Poemage that probe more deeply into the literary concepts discussed in this paper, as well as into new concepts such as sonic depth and the role of technology in promoting creativity.

9 Reflections

Working with poetry scholars has been a delightful, and challenging, process. Here we reflect on some of the issues that we believe make literary studies different from other, more traditional visualization problem domains. We offer a number of insights we gained as visualization designers and provide several suggestions for future work.

9.1 Breaking Convention

This research challenged us to embrace concepts that visualization conventions tell us to do otherwise, specifically, ambiguity and visual clutter. In the field of visualization, avoiding ambiguity is the norm. In this research, however, our collaborators led us to regard ambiguity as a fundamental source of insight. While Poemage includes some features that allow users to explore ambiguity within the data, we plan to investigate this topic in greater detail in our future work.

This work also challenged us to embrace a degree of visual clutter. The *beautiful mess* appends established visualization principles that value clarity and readability. This messy view, however, was consistently one of our collaborators' favorites, and it revealed one of the more important poetic insights of this work, shown in Figure 4 . Our collaborators told us they would not have made this insight without the beautiful mess. In an interview, one of our collaborators commented that the beautiful mess was completely representative of what she and other poets seek to understand in a poem, namely how the constituent parts of a poem work together to form complex meaning. This collaborator also reported that in times of feeling overwhelmed by the technology, she turned to the beautiful mess to ground and re-energize herself. We wonder, however, if there is a degree of novelty in the beautiful mess that may wear off in time — we plan to revisit the utility of this view in the future.

In general, coming to terms with how quickly the visualizations became cluttered, and *not* restricting the tool to avoid such clutter, was a challenge for us as visualization designers. To our collaborators, this clutter, which they identified more as *chaos*, was inviting and energizing and was a space that they felt very comfortable exploring. Similarly, our collaborators' excitement about ambiguity as an aspect of the data that enhances meaning, rather than clouds it, caused us to reconsider how to include it in Poemage.

This experience taught us to be willing to put some of our own design principles aside and to be open to experimenting with unconventional visualization if explicitly, or implicitly, requested. Doing so led us to include features that our collaborators were genuinely excited about and also helped us to better understand the problem space. Throughout this research, our collaborators actively challenged and probed their resistance to integrating technology into their practices in the hopes of advancing their research, and we as visualization designers learned to do the same. This is a lesson that we plan to bring to future collaborations, and we encourage others to do the same.

9.2 A Screwmeneutic Approach

Within the digital humanities, an adventurous wave of research rejects the notion of using computation to solve a text or to verify existing hypotheses, and instead focuses on using computers to further literary criticism, to generate an indefinite number of unique and sometimes radical interpretations and to guarantee continued meaning making. Concepts of text *deformation* [41] and *tamperings* [33] have energized members of the literary criticism community and have motivated a somewhat informal branch of text interpretation delightfully termed *screwmeneutics*. This term comes from an influential paper by Stephen Ramsay entitled “The Hermeneutics of Screwing Around...” [47]. Tools created with a screwmeneutics sensibility encourage a certain amount of playfulness and creativity from their users. We embraced this philosophy whole-heartedly throughout this design study, from our approach to conducting research to the design of technology probes and Poemage. Our validation results in Section 8 explore a range of possible outcomes on the part of our collaborators that can emerge when poets are given a tool that supports free-form exploration, browsing, and play.

Because a significant component of this research was investigating the role and impact of technology on the experience of close reading, and because our collaborators gained so much from this aspect of the design study process, we wanted to allow users to conduct similar investigations using our tool. In addition to being a culmination of our research findings, we see the various components of Poemage additionally as technology probes for end-users — opportunities to explore the impact of technology on their individual reading of a poem.

9.3 Measuring success

An underlying challenge throughout this design study has been determining how to measure our success. We attribute this to several different complications. First and foremost, the range of valid interpretations makes comparing against ground truths fairly unproductive. In terms of proper evaluation, our research fits in the *evaluating visual data analysis and reasoning* scenario [34], which encourages insight-based evaluations. In our case, in which gathering insights is a fundamental component of our collaborators’ research, this method seems less indicative of success than it perhaps might be for a different domain. Therefore, in addition to highlighting specific insights gained using our tool, we also evaluate new *kinds* of insights and how the insight gathering process may have changed using Poemage, which connects back to our investigation of the role and disruptive impact of technology on close reading.

In reflecting on the outcome of this research, the question arises: could randomly selected sets of words yield equally valuable and equally abundant insights? Our collaborators have hinted that the answer to this question might be *yes*, which leads to a second question: what is the role of technology in a creative pursuit like poetry? We are continuing to explore these ideas with our collaborators and are designing user studies to test our hypotheses.

In a similar vein, we acknowledge that the findings and contributions presented in this paper place a stronger emphasis on the role and impact of technology on poetry scholarship than on the support of close reading with Poemage. Although we are continuing to improve Poemage, we believe that developing a deep understanding of the intersection of technology and the humanities is fundamental to creating truly effective tools supporting a broad range of literary studies practices — extending beyond close reading.

Finally, in this research, *pleasure* and *enjoyment* are productive research outcomes and play an important role in guiding exploration [36] [11] [47]. One consequence of this, and as we found through our validation, creating a visually pleasurable research environment goes beyond general aesthetics, encouraging richer exploration and greatly increasing the overall efficacy of the tool. In reflection, we call for an adapted set of evaluation guidelines for conducting research in the humanities, and specifically in literary studies. We believe that such guidelines could extend to other arts as well.

10 Conclusions

We present a two-year design study exploring the role and influence of technology on the close reading of a poem via an investigation of sound and linguistic devices in poetry. The results of our research include a problem characterization and data abstraction of the use of sound in poetry, as well as Poemage, a visualization tool for interactively exploring the sonic topology of a poem. The design of Poemage was heavily informed by a series of technology probes iteratively developed with our collaborators. We validate our design through several case studies that highlight the kinds of insights gained using our tool and illustrate the disruptive impact that technology can have on poetry research. In the immediate future, we plan to address a current limitation of Poemage, which is the specification of interesting rhyme patterns by a poet. We plan to explore machine-learning algorithms that would enable a more free-form specification of rhyme rules through user-driven selection of words.

In our undertaking to visualize poetic sound, this project participates in several persistent lines of critical inquiry. The visualizations could contribute to discussions regarding concrete or *synesthetic’* poetry, which deliberately engage cross-modal perception [5, 31]; regarding reading and writing about text and image and about alphabetic signs as a basic digital technology that operates, already, on visual and aural/oral registers [5, 29]; or regarding how electronic digital (re)mediation further complicates semiotic and phenomenological questions through coding languages, hardware, and interface [38, 28, 22, 31]; etc. In future papers, we will articulate Poemage’s theoretical, literary positioning more fully. For now, we note Johanna Drucker’s observation that poetic texts activate both sonic and visual elements as “intersecting codes” [21], which can create experiences that are synesthetic in the etymological sense of perceiving together, or at the same time [31], and thereby lead to fuller aesthetic and interpretive engagement with poems. The computational visualization of poems promises to exploit this engagement in rich ways. Since in Poemage the text of the original poem remains always in view, rendering its sonic patterns as visual forms does not diminish sonic pleasure readers might experience apart from visualizations, but it has the strong potential to enhance it.

Acknowledgments

We would like to thank Craig Dworkin, Amanda Hurtado, and Michael Gleicher for their contributions to this work. We would also like to thank our reviewers for their feedback in improving this manuscript. This work was funded in part by NSF grant IIS-1350896 and NEH grant HD-229002.

References

[1] gicentre utilities. http://www.gicentre.net/utils/.

[2] A. Abdul-Rahman, J. Lein, K. Coles, E. Maguire, M. Meyer, M. Wynne, C. R. Johnson, A. Trefethen, and M. Chen. Rule-based visual mappings–with a case study on poetry visualization. In *Computer Graphics Forum*, volume 32, pages 381–390. Wiley Online Library, 2013.

[3] B. Alper, N. H. Riche, G. Ramos, and M. Czerwinski. Design study of linesets, a novel set visualization technique. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2259–2267, 2011.

[4] M. Bastian, S. Heymann, M. Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. *ICWSM*, 8:361–362, 2009.

[5] A. S. Bessa. Sound as subject: Augusto de campos’s poetamenos. In M. Perloff and C. Dworkin, editors, *The Sound of Poetry/The Poetry of Sound*, pages 237–248. University of Chicago Press, 2009.

[6] R. Buchanan. Wicked problems in design thinking. *Design Issues*, 8(2):pp. 5–21, 1992.

[7] H. Byelas and A. Telea. Visualizing metrics on areas of interest in software architecture diagrams. In *Visualization Symposium, 2009. PacificVis' 09. IEEE Pacific*, pages 33–40. IEEE, 2009.

[8] M. Chaturvedi, G. Gannod, L. Mandell, H. Armstrong, and E. Hodgson. Myopia: A visualization tool in support of close reading. *Digital Humanities*, 2, 2012.

[9] T. Clement. Distant listening or playing visualizations pleasantly with the eyes and ears. *Digital Studies/Le champ numérique*, 3(2), 2012.

[10] T. Clement, D. Tcheng, L. Auvil, B. Capitanu, and M. Monroe. Sounding for meaning: using theories of knowledge representation to analyze aural patterns in texts. *Digital Humanities Quarterly*, 7, 2013.

[11] K. Coles. Slippage, spillage, pillage, bliss: Close reading, uncertainty, and machines. *Wester Humanities Review*, pages 39–65, Fall 2014.

[12] K. Coles and J. Lein. Turbulence and temporality: (re)visualizing poetic time. things my computer taught me about poems. In *MLA2014. Chicago, IL. Jan. 2014.*

[13] C. Collins. Docuburst: Document content visualization using language structure. In *Proceedings of IEEE Symposium on Information Visualization, Poster Compendium*, 2006.

[14] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1009–1016, 2009.

[15] C. Collins, F. B. Viegas, and M. Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 91–98. IEEE, 2009.

[16] M. Correll, M. Witmore, and M. Gleicher. Exploring collections of tagged text for literary scholarship. In *Computer Graphics Forum*, volume 30, pages 731–740. Wiley Online Library, 2011.

[17] W. Cui, H. Zhou, H. Qu, P. C. Wong, and X. Li. Geometry-based edge clustering for graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1277–1284, 2008.

[18] P. DeCamp, A. Frid-Jimenez, J. Guiness, and D. Roy. Gist icons: Seeing meaning in large bodies of literature. In *Proceedings of IEEE Symposium on Information Visualization*, 2005.

[19] K. Dinkla, M. J. van Kreveld, B. Speckmann, and M. A. Westenberg. Kelp diagrams: Point set membership visualization. In *Computer Graphics Forum*, volume 31, pages 875–884. Wiley Online Library, 2012.

[20] A. Don, E. Zheleva, M. Gregory, S. Tarkan, L. Auvil, T. Clement, B. Shneiderman, and C. Plaisant. Discovering interesting usage patterns in text collections: integrating text mining with visualization. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 213–222. ACM, 2007.

[21] J. Drucker. Not sound. In M. Perloff and C. Dworkin, editors, *The Sound of Poetry/The Poetry of Sound*, pages 237–248. University of Chicago Press, 2009.

[22] J. Drucker. Humanities approaches to graphical display. *Digital Humanities Quarterly*, 5(1):1–21, 2011.

[23] J. Ellson, E. Gansner, E. Koutsofios, S. North, and G. Woodhull. Graphviz. *URL: http://www. research. att. com/sw/tools/graphviz*, 1998.

[24] O. Ersoy, C. Hurter, F. V. Paulovich, G. Cantareiro, and A. Telea. Skeleton-based edge bundling for graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2364–2373, 2011.

[25] J.-D. Fekete and N. Dufournaud. Compus: visualization and analysis of structured documents for understanding social life in the 16th century. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 47–55. ACM, 2000.

[26] S. Fish. Mind your ps and bs: the digital humanities and interpretation. *New York Times*, 23(1), 2012.

[27] B. Fry and C. Reas. Processing. org. *Processing. org*, 2010.

[28] N. K. Hayles. Translating media: Why we should rethink textuality. *The Yale Journal of Criticism*, 16(2):263–290, 2003.

[29] N. K. Hayles. Print is flat, code is deep: The importance of media-specific analysis. *Poetics Today*, 25(1):67–90, 2004.

[30] J. Heer and D. Boyd. Vizster: Visualizing online social networks. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 32–39. IEEE, 2005.

[31] P. Hertz. Synesthetic art—an imaginary number? *Leonardo*, 32(5):399–404, 1999.

[32] H. Hutchinson, W. Mackay, B. Westerlund, B. B. Bederson, A. Druin, C. Plaisant, M. Beaudouin-Lafon, S. Conversy, H. Evans, H. Hansen, et al. Technology probes: inspiring design for and with families. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 17–24. ACM, 2003.

[33] E. Irizarry. Tampering with the text to increase awareness of poetry's art.(theory and practice with a hispanic perspective). *Literary and linguistic computing*, 11(4):155–162, 1996.

[34] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Seven guiding scenarios for information visualization evaluation. 2011.

[35] A. Lambert, R. Bourqui, and D. Auber. Winding roads: Routing edges into bundles. In *Computer Graphics Forum*, volume 29, pages 853–862. Wiley Online Library, 2010.

[36] J. Lein. Sounding the surfaces: Computers, context, and poetic consequence. *Wester Humanities Review*, pages 84–109, Fall 2014.

[37] E. Lieberman, J.-B. Michel, J. Jackson, T. Tang, and M. A. Nowak. Quantifying the evolutionary dynamics of language. *Nature*, 449(7163):713–716, 2007.

[38] D. Lopes. *A philosophy of computer art*. Routledge, 2009.

[39] S.-J. Luo, C.-L. Liu, B.-Y. Chen, and K.-L. Ma. Ambiguity-free edge-bundling for interactive graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 18(5):810–821, 2012.

[40] N. McCurdy, V. Srikumar, and M. Meyer. Rhymedesign: A tool for analyzing sonic devices in poetry. In *Proceedings of Computational Linguistics for Literature*, 2015.

[41] J. McGann and L. Samuels. Deformance and interpretation. *New Literary History*, 30(1):25–56, 1999.

[42] W. Meulemans, N. H. Riche, B. Speckmann, B. Alper, and T. Dwyer. Kelpfusion: A hybrid set visualization technique. *Visualization and Computer Graphics, IEEE Transactions on*, 19(11):1846–1858, 2013.

[43] A. Muralidharan and M. A. Hearst. Supporting exploratory text analysis in literature study. *Literary and linguistic computing*, 28(2):283–295, 2013.

[44] M. Nöllenburg. An improved algorithm for the metro-line crossing minimization problem. In *Graph Drawing*, pages 381–392. Springer, 2010.

[45] W. B. Paley. Textarc: Showing word frequency and distribution in text. In *Poster presented at IEEE Symposium on Information Visualization*, volume 2002, 2002.

[46] S. Pupyrev, L. Nachmanson, S. Bereg, and A. E. Holroyd. Edge routing with ordered bundles. In *Graph Drawing*, pages 136–147. Springer, 2012.

[47] S. Ramsay. The hermeneutics of screwing around; or what you do with a million books. *Stephen Ramsay*, 17:103, 2010.

[48] S. Ramsay. *Reading machines: Toward an algorithmic criticism*. University of Illinois Press, 2011.

[49] H. W. Rittel and M. M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4(2):155–169, 1973.

[50] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2431–2440, 2012.

[51] B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, pages 1–7. ACM, 2006.

[52] J. Stasko, C. Görg, and Z. Liu. Jigsaw: supporting investigative analysis through interactive visualization. *Information visualization*, 7(2):118–132, 2008.

[53] M. Suri. Don't expect math to make sense. *The New York Times.*

[54] F. Van Ham, M. Wattenberg, and F. B. Viégas. Mapping text with phrase nets. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1169–1176, 2009.

[55] F. B. Viegas, M. Wattenberg, and J. Feinberg. Participatory visualization with wordle. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1137–1144, 2009.

[56] M. Wattenberg. Arc diagrams: Visualizing structure in strings. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 110–116. IEEE, 2002.

[57] M. Wattenberg and F. B. Viégas. The word tree, an interactive visual concordance. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1221–1228, 2008.

[58] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Information Visualization, 1995. Proceedings.*, pages 51–58. IEEE, 1995.