¡¡¡¡¡¡¡ HEAD ======= ¿¿¿¿¿¿¿ upstream/master ¡¡¡¡¡¡¡ HEAD ======= ¿¿¿¿¿¿¿ upstream/master

¡¡¡¡¡¡¡ HEAD ======= ¿¿¿¿¿¿¿ upstream/master ¡¡¡¡¡¡¡ HEAD ======= ¿¿¿¿¿¿¿ upstream/master

An Evaluation of Poemage

Daniel Schroeder, Will Sims, and Bungo Takahashi

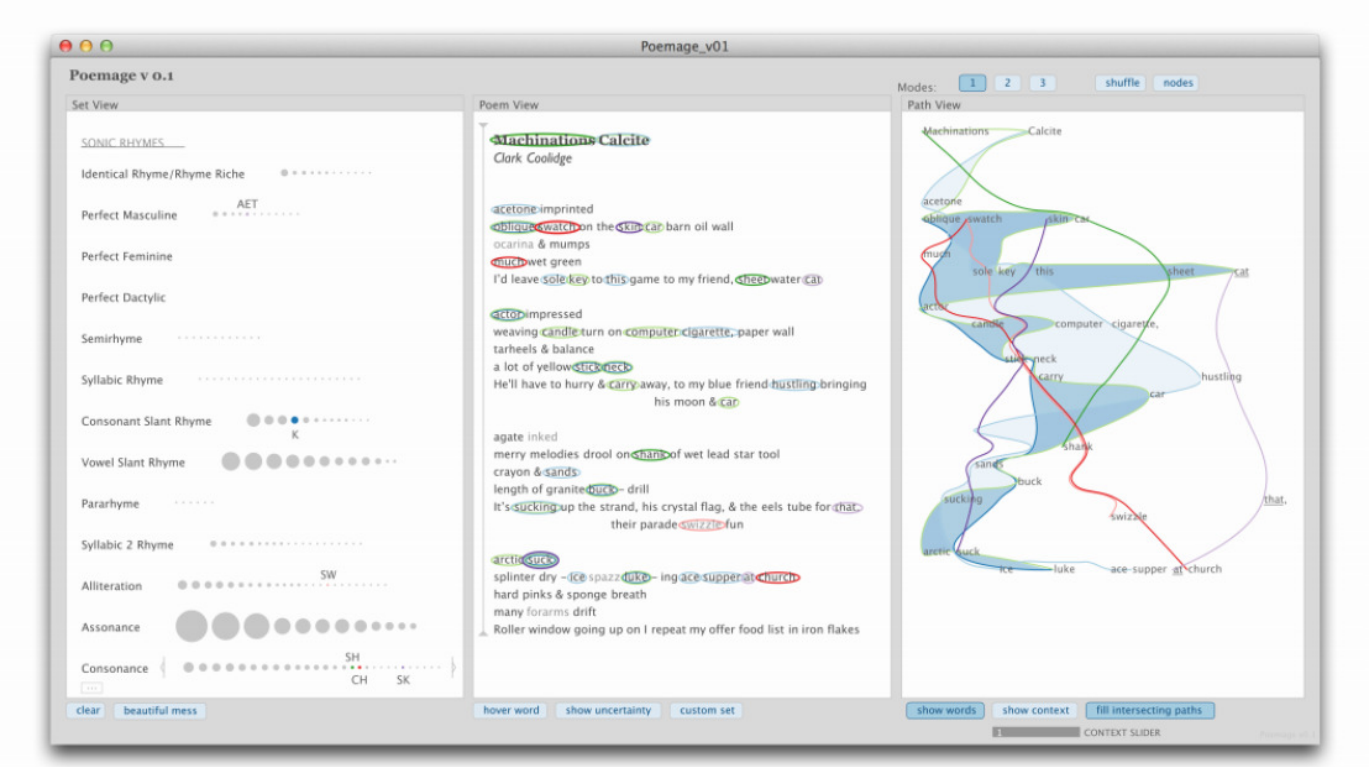

Fig. 1. The Poemage interface comprises three linked views: (left) the set view allows users to browse sets of words linked through sonic and linguistic resemblances; (middle) the poem view allows users to explore sonically linked words directly via the text; (right) the path view shows the sonic topology of a poem. [5]

Abstract—This paper is a visualization evaluation of the program Poemage, created by Nina McCurdy, Julie Lein, Katharine Coles, and Miriah Meyer from the University of Utah. We will evaluate the effectiveness of this visualization as well as suggest future implementations that could enhance the application.

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

1 Introduction

1.1 What is Poemage

Poemage is a sonic topology visualization tool for poetry. The project was designed and created by a team of computer scientists and poets at the University of Utah. Poemage contains three types of data which are poem space, rhyme sets, and sonic topology. In Poemage, words have their own locations defined by the words position in a line and the lines position in the poem space. In addition to that, words belong to rhyme sets which are calculated by Poemages rhyming scheme, and there are 24 different kinds of rhyme types in the current version of Poemage. Then, sonic topology is calculated based on those sets of words classified in rhyme sets. Using those three attributes, Poemage visualizes sonic topology of poems and help users understand how words in a poem relate to one another.

1.2 Problem

The problem behind the project was how the computer can contribute to close reading. In general, there are two types of reading: distant reading and close reading. In distant reading, *"It aims to generate an abstract view by shifting from observing textual content to visualizing global features of a single or of multiple text(s)"*. [3] On the contrary, in close reading, it includes analysis on individuals, events, and ideas, their development and interaction, used words and phrases, text structure and style, and argument patterns. [3] As we know, the computer can process a lot of data, and textual data is not an exception, which means it is good at distant reading. However, it has been a question if the computer can be used for close reading, because it requires analysis on a single text from a lot of viewpoints. Having said that, the project team from the University of Utah decided to create a tool that focuses on one

- *Daniel Schroeder is a computer science student at Oregon State University. E-mail: schrodan@oregonstate.edu.*
- *BUNGO Takahashi is a computer science student at Oregon State University. E-mail: takahasb@oregonstate.edu.*
- *Will Sims is a computer science student at Oregon State University. E-mail: simsw@oregonstate.edu.*

of the aspects in close reading, which is sound, so that the computer can contribute to that type of reading.

1.3 Purpose and Importance

As explained earlier, close reading requires analysis on a single text from different points of view, and sound is one of them. Especially in poetry, the interaction of sonic devices between words is complex, which makes it hard for poets to analyze rhyme efficiently. The poemage project addresses this difficulty by visualizing sonic topology so as to generate productive meaning from poems in terms of sound, and it helps users understand relationships and interactions between words more effectively. This visualization is important because it is difficult for a human to capture all of the relationships between words and to analyze them by hand. In addition to that, the visualization gives users new perspectives which they had not realized. Therefore, it is beneficial to visualize sound topology thoroughly by utilizing the software.

2 Related Work

Close reading through computational means is a very difficult problem to tackle due to its large problem set and complexity by nature. Most design decisions for computational close reading tend to focus on a specific area of interest within close reading, for example sound or rhyme. Nina McCurdy from The University of Utah wrote: "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" when describing the complexity of text analysis [5].

There have been numerous previous projects trying to tackle close reading via visualization and computation like E-Margin [4], Serendip [1], and Myopia [2]. The Poemage team took a slightly different course during their development by focusing on sounds and creating a program that could "automatically sonify a poem" [5]. All previous projects address the problem of close reading as an interpretation of text at the level of individual words in order to interpret and examine literary context. A large issue with addressing this problem is that it is usually done in an academic setting where someone is trying to "teach" the literature, and hand written evaluations tend to be messy and difficult to understand. These close reading visualization projects attempt to help visualize textual interpretations so that it can be taught and evaluated easily and effectively.

We wanted to branch off from the Poemage style of using drawn ellipses to highlight words of interest and search for other methods of text highlighting that may increase the effectiveness of the visualization. We sought to highlight individual syllables or letters that applied to the filtered rhyme scheme or rhyme rule rather than circling the entire word. There has been extensive research on text highlighting techniques conducted by the information visualization community. A paper from IEEE Transactions on Visualization and Computer Graphics states that "Text highlighting is important in any scenario where close reading (sequential word-by-word reading) is required and text annotations exist, that should be made accessible to the reader" [7]. We wanted to explore the possibilities of using background coloring to target different parts of a single word to beautify the user interface as a whole and enhance the effectiveness of the Poemage program. By using different background colors used to target specific letters and syllables, the user will be able to clearly identify the sonic devices being targeted and be able to see where in the word it is present. The issue of where in the word the sonic device is located is not expressed through the use of drawn ellipses in the current version of Poemage, and users become overpowered with clutter and complexity when multiple filters are applied.

3 Method

In the methods section we discuss how we carried out the usability evaluation of Poemage and the evalution of Poemage with different languages.

3.1 Method Overview

Poemage is a tool that was created for close reading which is primarily used in education. We wanted to build on this application by making assessing how well it meets the needs of students and people that speak languages other than English. In order to accomplish this, we evaluated the effectiveness of Poemage while using different languages such as Spanish, and also conducted a usability evaluation to evaluate the ease of use of the application.

3.2 Usability Evaluation

According to Nielsen, Usability is a quality attribute that assesses how easy user interfaces are to use. The word **usability** refers to methods for improving ease-of-use during the design process [6]. Since it very likely Poemage will be used in an education setting, it is critical that students are able to understand how to use the application. Usability is especially important because Poemage is only useful when it is more efficient than analyzing a poem by hand. We conducted a cognitive walkthrough before creating our tasks for the usability evaluation so that we could find areas of interest that we would like to test with users.

The cognitive walkthrough showed that some of the problem areas may include the ambiguity of the modes section, overlapping text when selecting all parts of a sonic rhyme. It was also unclear whether or not there are more sonic rhymes below Phonetic Assonance in the set view, and it is difficult to see which syllables in a word are actually making the sound that rhymes with other words. We used these findings from the cognitive walkthrough to construct the tasks that users would perform for usability testing.

3.3 Tasks, Metrics, and Users

We created two tasks for users to perform in order to evaluate the usability of Poemage. The first task was to view the poem Night by Louise Bogan and select EH and AY vowel slant rhymes. Parcicipants were then asked if they could explain what a vowel slant rhyme was from the output. Lastly, we instructed them to change the mode to 2 and explain what they believed had occurred from that change. The purpose of this task was to see if users understood how to select a specific syllable from the Vowel Slant Rhyme section and interpret the output. We also wanted to see if they understood the different modes in Poemage.

The second task was to view the poem EMILYDICKINSON_861. They were then instructed to select S from the single character visual rhyme. We also asked them how they interpreted the results that were displayed on the screen. The purpose of this task was to see if they knew how to select a poem from the list and scroll down in the set view in to find the visual rhymes section. We also wanted to see if they understood what a visual rhyme is and if they knew how to select a character from the list.

The users which were used for the evaluation of Poemage were obtained through convenience sampling. We recruited 5 participants (most of these were roommates) to carry out the tasks described earlier in this section. None of the users were familiar with the program, but most of them possessed basic technology skills.

The metrics that we measured when evaluating the usability of Poemage are task success, errors, and qualitative responses from users. We wanted to gather qualitative data to see if the user understands what Poemage is used for and whether or not they know the difference between the sonic and visual rhymes.

3.4 Poemage With Different Languages

English is the only language supported by Poemage, but we wanted to test it with a poem translated in Spanish to see if the algorithm would work for other latin languages. In order to evaluate this, we used The Road Not Taken by Robert Frost and translated it into Spanish using Google Translate. We also used [insert Spanish poem] to evaluate the effectiveness of Poemage when using a piece that was originally written in Spanish.

4 Enhancements

For our proposed enhancements, we want to enhance the overall usability and clarity of the Poemage user interface and provide additional

services such as on-hover abilities and multiple language support.

4.1 Colored Background Text Highlighting

The current instantiation of text highlighting in Poemage creates colored ellipses around words in the text that pertain to the given filters applied. This technique causes clutter in the middle viewer window when numerous filters are applied and does not give sufficient information about where the sonic device is located within the word itself. Our solution involves coloring the background behind the word or syllable with a soft, distinguishable color to highlight the part of the word that is being evaluated. Not only will this decrease the clutter in the viewer window, but it will also provide information about the sonic devices that the user was not previously available to.

As shown in the figure above, our implementation will allow for multiple background colors to be partitioned throughout a single word and clearly identify which syllable is being targeted by which filter. We feel as though this technique provides clearer and more precise analysis of the given text and provides user with a better understanding of the sonic devices present.

4.2 On-Hover Functionality and Text Spacing for Filters Window

The filters viewer window has some UI issues we wanted to address and improve upon, specifically the spacing of the text that name the syllable sounds and descriptions of the rhyme types.

The filter window has clickable circles of varying size that correlate to the instances each rhyme rule occurs in the text being analyzed. When one of these circles is clicked, text appears above or below the circle with letters describing the sound each sonic rule targets. When all the circles for a given sonic device are clicked, the texts tend to overlap and become illegible. We anted to implement a redesign of this text to allow padding around the letters and remove any chance of overlap or muddled text and improve readability by the user.

Poemage was developed by scholars and poets, which resulted in a design implementation meant for users that are avid in linguistics and poetic analysis. For novice users, there are a lot of features that are confusing and poorly explained. This is why we would like to add an on-hover feature on the filters list that gives the user a description of what the rhyme rules mean and the given characteristics of each filter (i.e. Masculine Rhyme or Assonance).

4.3 Arrow to Indicate Scrollable Filters Menu

The filters viewer window is a scrollable window that has multiple rhyme rule filters that are out of view when the program is initiated. The issue is that there is no indication that users can scroll to reveal more filters that are not currently in view. We suggest implementing a simple grey arrow at the bottom of the filters window that will inform users they have the ability to scroll down to access more filters. We thought about a way to fit all the filters into a single view so that no scrolling was necessary, but to maintain the tri-window interface and keep the readability, we decided that compacting filters closer together was not optimal. Rather, a simple arrow for indication would ensure users knew more filters were present seemed like the best improvement.

4.4 Remove Mode Buttons

On the top right corner of the Poemage interface, there are three mode buttons that shift the nodes in the path view window to remove whitespace or create a uniform spacing between nodes. We felt as though these buttons were not necessary and relatively confusing to novice users. The different modes are used to format the nodes according to different spacing settings to shift the structure of the path view window. To keep this interface user-friendly and as effective as possible for textual analysis, we plan to remove these mode buttons and render them unhelpful for the program as a whole.

4.5 Support for Multiple Languages

Our biggest improvement we want to perform is the support for multiple languages by the Poemage program. Currently, Poemage is implemented with rhyme rules and text parsing meant for Latin based languages and creates sets sonically related words that comply to rules of the English language. To create multiple language support, this would mean implementing an immense amount of code reconstruction and new functions that would correctly parse and categorize syllables in other languages. This would mean an entirely new code base that has language rules and parsing techniques that are able to recognize non-English sounds and letter combinations. Although difficult, we feel like this would be a very beneficial improvement to the Poemage program and extend the projects benefits to a broader audience.

5 Results and Performance

5.1 Performance Analysis

6 Conclusions and Future Work

Acknowledgments

The authors wish to thank Nina McCurdy and the faculty at The University of Utah that built the Poemage program as well as professor Eugene Zhang and teaching assistant Islam Al Musaly.

REFERENCES

[1] E. Alexander, J. Kohlmann, R. Valenza, M. Witmore, and M. Gleicher. Serendip: Topic model-driven visual exploration of text corpora. *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2014. doi: 10.1109/vast.2014.7042493

[2] M. Chaturvedi, G. Gannod, L. Mandell, H. Armstrong, and E. Hodgson. Myopia: A visualization tool in support of close reading. http://www.dh2012.uni-hamburg.de/conference/programme/abstracts/myopia-a-visualization-tool-in-support-of-close-reading.1.html.

[3] S. Janicke, G. Franzini, M. F. Cheema, and G. Scheuermann. On close and distant reading in digital humanities: A survey and future challenges. http://www.informatik.uni-leipzig.de/~stjaenicke/Survey.pdf.

[4] A. Kehoe and M. Gee. emargin: A collaborative textual annotation tool. *Ariadne*, 71, 2013.

[5] N. Mccurdy, J. Lein, K. Coles, and M. Meyer. Poemage: Visualizing the sonic topology of a poem. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):439448, 2016. doi: 10.1109/tvcg.2015.2467811

[6] J. Nielson. Usability 101: Introduction to usability.

[7] H. Strobelt, D. Oelke, B. C. Kwon, T. Schreck, and H. Pfister. Guidelines for effective usage of text highlighting techniques. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):489498, 2016. doi: 10.1109/tvcg.2015.2467759