VerseVis: Visualization of Spoken Features in Poetry

Austin Myers
University of Maryland
College Park, MD 20740
amyers@umd.edu

Leslie Milton
University of Maryland
College Park, MD 20740
lmilton@umd.edu

Christine Lu
University of Maryland
College Park, MD 20740
clu@cs.umd.edu

Abstract—**The exploration and analysis of literary corpora is a difficult task. Previous approaches to this problem focused on mining data directly from text. However, these solutions do not aid researchers who are interested in learning spoken features of the text, which play an important role in poetic works. VerseVis is a text visualization tool that gives users the ability to identify interesting text patterns within literary corpora. The current implementation focuses on various ways to explore spoken language features of text, like phonemic expressions and stresses. Feedback from usability test participants show that representing phonemes and accents through colors and height is a useful tool for getting an insight on a body of text.**

Keywords-component; formatting, text visualization, information visualization, phoneme analysis

I. Introduction

English scholars and computer scientists have recently developed an interested in computer aided literary analysis. However, the high dimensionality of a body of text, compounded by the many possible combinations of phrases and words makes computational analysis a difficult task. Furthermore, to derive a semantic understanding of the text sample, it becomes necessary to utilize methods in natural language processing. The solution is to reorganize the data into a succinct symbolic representation. Ideally, the results produced by the visualization help users extract and identify relationships between words, structure, and document flow.

There have been several recent approaches in this area. FeatureLens [4] visualized a text collection at several levels to find repetitions and other text patterns. By using the program, the user could identify phrases that used similar words and syntax. Compus [5] explored visualizations for a corpus of French manuscripts. It focused on finding regularities and discrepancies by allowing the user to restructure the visualization with selected elements. Finally, Exploring Erotics [8] visualized a series of Emily Dickinson poems to extract a holistic impression of the risque quality. It does so by analyzing the indicators of erotics, document states, and rhetoric.

The previous works identified possible relations between interesting text patterns, but tended to focus on features like the frequency of words, the distribution of parts of speech, or the comparisons between different bodies of text. However, these qualities of the texts do not explore a further field for variation: the vocal quality of poetry.

VerseVis, introduces a set of visualization functions that allow users to explore and display fundamental properties of poetic verse: rhythm, meter, alliteration, consonance, and assonance. Our project focuses on exploring the various verbal and spoken qualities that are largely unique to poetry.

To support discovery of new insights, VerseVis provides novel methods of exploration. The user can filter results to view poems holistically or zoom in to receive details on demand by viewing stanzas, phrases, or words. A search can be applied across the data set to highlight relevant sections of text. The graph can also be partitioned by colors to represent additional features, like parts of speech, word index, and phoneme data.

The following sections describe motivation, related works, poetic elements, data description, design issues, features of VerseVis, application examples, evaluation and suggestions for future work in this area.

II. Motivation

The motivation for VerseVis is based on the poetic works of Gertrude Stein. Stein's writing stands out from the collection of poetic texts because it often makes use of repetition and tautologies (cyclical phrases). Furthermore, Stein's work also plays heavily on the phonemic content of the lines to evoke specific emotions from the reader.

Gertrude Stein once wrote: “Before the flower of friendship faded, friendship faded.” When read aloud, this line exhibits iambic pentameter (a specific type of cadence). Through emphasis and the consonance of the “f” sound, the line comes across as weary and tired. From the definitions of the words themselves and from the sentence structure and syntax, these characteristics are not immediately obvious. A closer examination at the role of sound in text can lead to a greater understanding of its effects and influences.

Sound is clearly a large part of the poetic arts. Early poems were passed down orally, and the writers used

Sponsored by Tanya Clement, University of Maryland.

prosodic elements to aid memorization. Since it has such a great effect on how humans interpret even the shortest lines of text, we created VerseVis to generate visualizations that aid in the analysis of the unique attributes and qualities of poetry. Ultimately, we hope to bring about a greater understanding of, not only the works of Gertrude Stein, but also of this field.

III. Related Work

Visualization scientists have developed various techniques to evaluate different kinds of textual data. Highlighting text and relationships in a document and analyzing color-coded text can provide valuable insights to the structure and tone of a textual document.

TOPIC ISLANDS is a visualization system prototype that seeks to reduce the amount of time a user should spend on reading long articles [1]. In some cases, users want to know the most important information without reading the entire document. This system is used to browse a document, generate fuzzy outlines, summarize text according to user interests and by levels of detail, query text content, and provide summaries of topic evolution. *TOPIC ISLANDS* applies wavelet transforms to a customized digital signal constructed from words in a document.

Weber [2] suggests an approach to text visualization by the use of color codes that display word classes. The colored words can provide some level of detail about the writing style, sentence structure, and the text genre. The color-code system follows the meaning of words according to western culture.

Users can understand data better by interacting with information that is viewed using different representations. Coordinated and Multiple Views (CMV) [3] is an exploratory visualization method that gives users the ability to explore such data. During the analysis of complex investigations, the user may need to consider various scenarios to compare visualizations generated from the same or multiple datasets. As a result, a comprehensive exploratory tool with intuitive functions should be used. CMV can help the user discovery insightful relationships and facts from their data.

Text mining results are complex structures that suggest interesting patterns for researchers to analyze. However, text mining algorithms return a large number of patterns that prove to be difficult to interpret. As a result, *FeatureLens* [4] was designed to make text mining results more comprehensible. It allows users to analyze text mining results through visual exploration. The discovery process is guided by the concepts of frequent expressions, words, and closed item sets of n-grams. A similar text visualization tool, *Compus* [5], explores corpora of structured XML documents. This tool finds regularities and discrepancies in historical documents as well as plays and literary corpora. *Emily* [8] is another interesting text visualization tool that visualizes free-flowing natural language text to give users the ability to analyze the text without leaving the visualization tool.

Our project is mainly inspired by the *Compus*, *FeatureLens*, and *Emily* text visualization tools. These tools collectively provide text visualization features that will give analysts the ability to derive useful insights.

IV. Poetic Elements

A. Overview

A specific subset of written text is poetry. It is differentiable from prose and journalistic writing by certain features; poetry exhibits and utilizes qualities like rhythm, meter, rhyme, and alliteration. While there are no blanket purposes, poetry can be used to explore ideas, evoke emotions, and sometimes, tell a story.

B. Rhythm and Meter

TABLE I. Common Metrical Feet in Poetry

Name	***Length (Syllables)***	***Stressed Syllable(s)***	***Example***
Iamb	2	2	Attempt
Trochee	2	1	Odor
Dactyl	3	1	Mockingbird
Anapest	3	3	Halloween
Spondee	2	1,2	Big Deal

The differences between poetry and other text are largely owed to its use of stresses and accents to produce a rhythmic effect in oral dictation. For example, certain syllables are read with greater emphasis and force, while other syllables are held for longer durations. These patterns, through repetitive use in a stanza, bring out a cadence and musical quality in a reading. Because of this emphasis on rhythm, poetry possesses a unique ability to bypass conventional rules in grammar, syntax, and word order. The study of such rhythmic patterns is known as prosody. The most commonly used pattern units recognized in poetry are described Table I.

C. Rhyme

Another important property of poetry is rhyme. Rhyme, the repetition of similar sounds in two or more words, is a common method authors employ to improve the flow of a sentence.

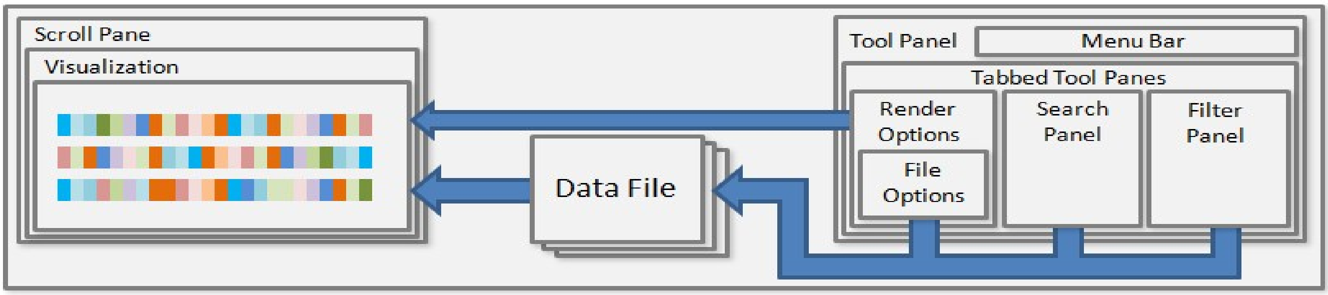

Fig. 1. VerseVis Architecture

D. *Alliteration: Consonance, and Assonance*

Similar to rhyme, alliteration is used to facilitate flow as well as emphasize a particular point of the piece of writing. It is defined as the repetition of the same sound in close succession, generally following along the natural cadence of the sentence, existing at the natural stresses of the poem's meter. It is further defined by the type of sound it repeats: consonance, much as the name suggests, applies to the repetition of consonant sounds, while assonance is concerned with the reiteration of vowel sounds.

Consonance: The **s**ilken **s**ad un**c**ertain ru**st**ling
- Edgar Allen Poe

Assonance: The cr**u**mbling th**u**nder of seas
- Robert Louis Stevenson

The use and frequency of alliteration can provide information about both the poet and the time period from which it was written - for example, alliteration is commonly used in Old English (and other Germanic languages) poetry.

V. Data Description

The Mary Text-to-Speech System [6] is used to find desired visualization features. Mary is an multilingual, open-source Text-to-Speech synthesis system with four main parts; pre-processing, natural language processing (NLP), calculation of acoustic parameters, and synthesis.

Pre-processing includes numeral expansion, abbreviation expansion, and tokenization. Throughout this process, an internal XML structure is built around the input text. NLP is first responsible for the part of speech labeling and shallow parsing. After this step is complete, a lexicon look-up is performed. Next, intonation and phrase structure symbols are assigned using part of speech info, punctuation, and local syntactic information. The output from the NLP component is translated into an acoustic parameters file. This file can be viewed by MBROLA, an algorithm for speech synthesis, to view phone symbols. Finally, a synthesizer is used to create a new file using a phoneme string.

VI. Architecture Description

VerseVis loads documents into a hierarchical data structure using format parameters which designate the levels of the document hierarchy [Fig. 1]. VerseVis can read tab delimited variable files based on these format parameters, which allows users to provide custom columns of data.

Mary Text-to-Speech XML output can also be read into to a hierarchical structure. Data at each level of the hierarchy is linked sequentially so that VerseVis can easily iterate through the data at a specific level. This also allows VerseVis to change the columns and lines displayed in the visualization without restructuring the internal representation of the data. The hierarchical structure of documents also enables filters or searches to be placed on criteria at any level of the document hierarchy.

The VerseVis application is composed of three main components; the visualization, tools, and data [Fig. 1]. First, a document's data structures are encapsulated by a DataFile object which other modules of VerseVis can access. Rendering options in the tool box modify the methods used by the visualization model for rendering data. The file options, search, and filter settings modify the visibility and rendering attributes of the document data elements. Finally, the visualization modules are responsible for iterating through the visible elements of the documents that need to be rendered to the screen.

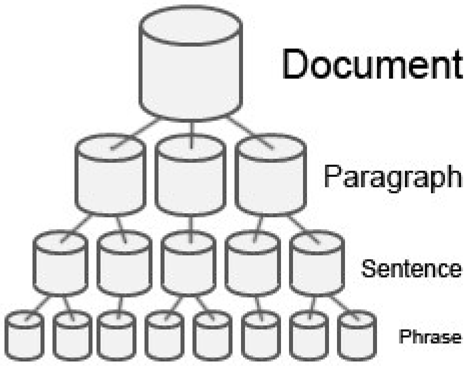

Fig 2. Hierarchical division of data

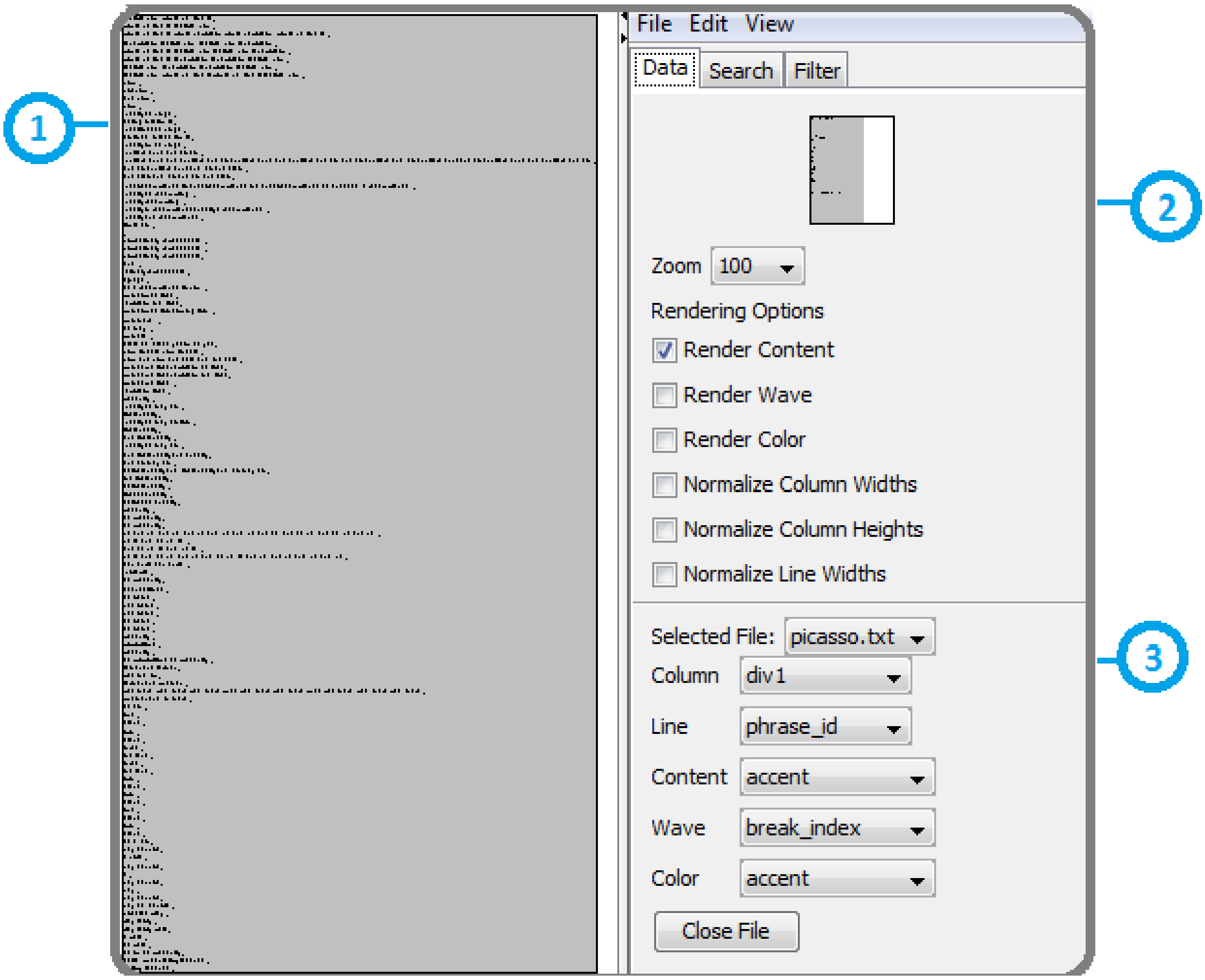

Fig 3. VerseVis display when text is first loaded
1. Visualization Display
2. Rendering Options Toolbox
3. Visualization Options Toolbox

VII. Interface Description

A. *Overview*

The interface is split into three main sections that can be seen in Figure 3: Visualization Display (1), Rendering Options Toolbox (2), and Visualization Options Toolbox (3).

B. *Data Input*

To start using VerseVis, the user provides a data file to the program by opening it from the File menu option. After the text file is opened, the information will be processed and stored in the program's internal data structure.

Once the data is loaded, the GUI's display updates simultaneously with three important changes in each of the main sections of the interface. In accordance with Schneiderman's "Overview first, zoom and filter, details on demand" [6] design principle, the initial display, once loaded, will show the entire body of text provided. Because of the differences in line length, the visualization will fill the width of the window with each line. Second, the Visualization Options Toolbox's "Selected File" drop down menu will automatically choose the most recent opened file. The pane will also expand to show additional visualization options that are now available. Finally, the rendering options toolbox will update with a thumbnail of the overall visualization.

When loading data to VerseVis, the user can open multiple files at a time. This allows comparisons between a series of independent poems and provides a means for evaluating trends. When multiple sets of data are loaded, again, the displays will be modified such that both visualizations for each text file tiled horizontally.

C. *Rendering Options Toolbox (2)*

The Rendering Options Toolbox allows the user to control visual aspects of the display. The toolbox's function is to control the visibility of the data in the display panel. The first control is labeled "zoom". Since the initial visualization provides an overview of the entire text, details in the data may not be immediately visible. Using the Rendering Options Toolbox, the user can adjust the scale of the display. To accommodate the changes, the display panel becomes scrollable.

There are also controls for visualization aspects: content, wave, and color. While the values and conditions of these variables are controlled in the Visualization Options Toolbox (3), each one can be individually removed or added to the visualization by the three "rendering" tools.

The following three checkboxes allow the user to normalize column width, column height, and line width, which may be a useful feature when comparing multiple documents.

D. *Visualization Options Toolbox (3)*

The first option available is the "Selected File" option. Since multiple data files may be added to the workspace and viewed simultaneously, users can independently select what attributes they would like to modify. Changes made to one display are not automatically applied to the second display.

The "columns" option selects the highest level in the visualization hierarchy; all the data gets partitioned into sections based on this user selected feature. These sections are then tiled horizontally in the display pane.
Similarly, the second option "line" allows the user to further partition the horizontal sections. These results are then tiled vertically within the columns blocks.

The Visualization Options Toolbox also contains tools that allow the user to control variables in the imaging. The display produced is a line graph whose attributes are declared by the following options: content, wave, and color. "Content" controls the label of the data point. The position of the data points remain in the order that they are encountered in the text file. The "Wave" option controls the physical locale of the line, where greater height indicates stronger value. As for the final option, the "color" will modify the color of the section of the graph based on the selected feature.

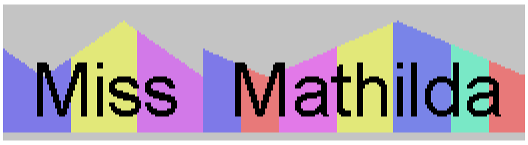

Fig. 4. Example of text display.
The height represents stress on the word, color represents phoneme.

E. *Other Functions*

VerseVis also contains a "Search" function that allows the user to search for a specific word. This word is highlighted within the entire body of text.

Finally, there is a filter function that allows the user to remove a word from the display. Additionally, the user can filter out visualization elements from everything but the last word in each line. This aids the user in identifying rhyming phrases.

VIII. Design Issues

The goal of VerseVis is to explore and display some of the sounded properties of poetry: rhythm, meter, alliteration, consonance, and assonance. For this exploration, the phonetic data can be used to identify similar verbal qualities from a body of text. However, because of the nature of this problem, each phoneme in each poem has to be rendered individually. Advanced preprocessing is not possible for this specific task, and the display can't be condensed to a compact smaller display without losing the feature intended for user exploration.

This compounded with the high dimensionality of data and the sheer volume of information available means that the program's performance is sensitive to the size of the data set. While there is no limitation on the size of the data set, an epic like The Iliad will reduce responsiveness of the application and lower performance.

Another concern is ensuring that the finished product would be accessible to our target audience, Tanya Clement and her research group. Because of the differences in operating systems and varying levels of experience, we decided to implement VerseVis with Java JRE 6. The GUI was built using Java Swing and AWT components

IX. Application Examples

Some users may need to analyze patterns that occur in text based on parts-of-speech. This task can easily be accomplished by changing the default settings. The text and color settings should be set to *part_of_speech* and *Render Color* respectively. Now the user has the ability to visualize all parts-of-speech in the document by color. If the user is only concerned with the colors in the visualization, the *Render Text* check box can be deselected. This feature shows various trends that a writer follows when creating textual documents.

For example, when analyzing the first 8 lines of Gertrude Stein's *If I Told Him,* it is clear that there is repetition or the use of the same parts-of-speech in this body of text. Figure 5 illustrates the parts-of-speech visualization feature.

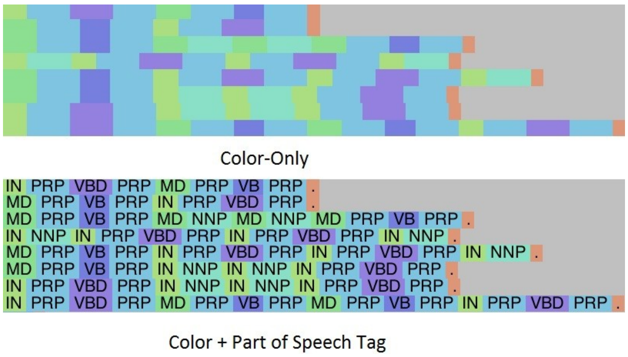

Fig. 5. Application Example
Application example showing color and part of speech features on a sample poem.

To give another example of an application, we imported Shakespeare's sonnet #1 (Fig. 6). In A, we first rendered phonemes by color. Then in B, we filtered the results to show only the last word of each line. This allows us to identify rhyming lines. The alternating colors of the last words of the first four lines indicate an ABAB rhyming pattern. Finally, in C, we render the wave with the accent feature. Again, by examining this result, we can see that it falls under the meter of iambic pentameter.

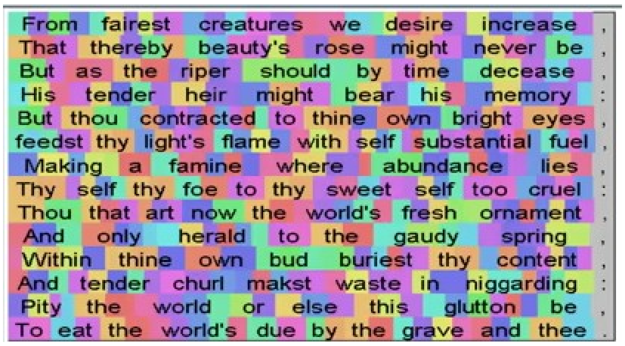

Fig 6.A: Colored Phonemes

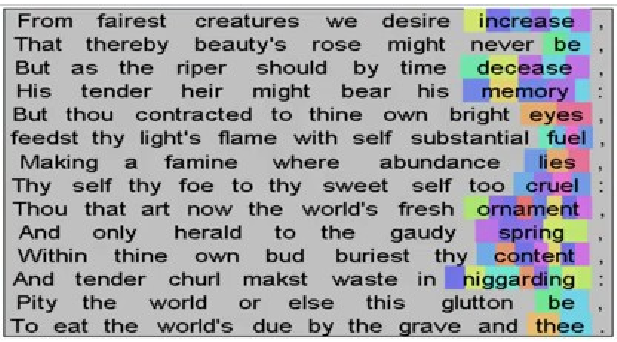

Fig. 6.B: Colored Phonemes – Filtered Results

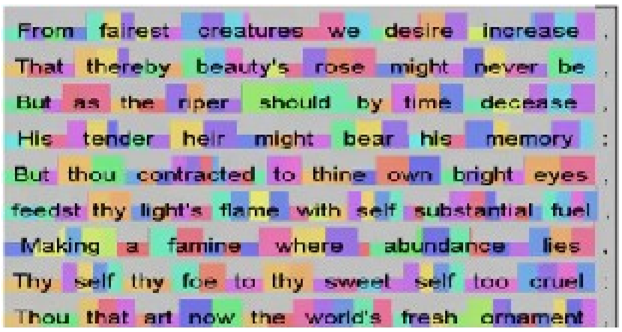

Fig. 6c. Accent Information

X. Evaluation

The team gathered insightful responses during the evaluation phase of VerseVis. This phase gave us the ability to access the system's quality and sophistication in a short time period by consolidating the viewpoints of experts and those that could potentially benefit from using the tool. We used a survey and interviews to identify the strengths and weaknesses of VerseVis to improve its usability.

The survey (See Appendix A for questions) was sent to graduate English students at the University of Maryland. The survey consisted of two portions. The first half of the survey asked questions about the responders' studies, how frequently they read poetry, how often they analyzed poetic elements of literary works, and how often they used data processing tools for analysis. The second half of the study was administered after the participants watched a demonstration video.

Eight English students responded to the survey sent to the department; seven of whom were Ph.D. students and one of whom was a student pursuing a M.A. degree. Figure 7 shows the responses to three questions from the survey.

An important insight that we found is that students indicated that they very rarely use data processing tools to analyze text. One student comments that "There aren't many English students or professors at UMD interested in text analysis." Though our project potentially addresses a niche problem, we believe that our tool can demonstrate usability in the field.

A list of the questions sent to each student is provided in the appendix. In summary, our findings showed that most participants found the color features the most interesting and useful part of the project. Other noteworthy, interesting characteristics included accents and emphasis indicators. On the other side of the spectrum, participants found the height and width adjustment features the least interesting.

Based on user input, we found that there is a demand for footnotes, definitions, and accents. There was also a suggestion that an enjambment and cesura feature could be included to identify puns. Most importantly, the use of interpretive options would help derive useful insights.

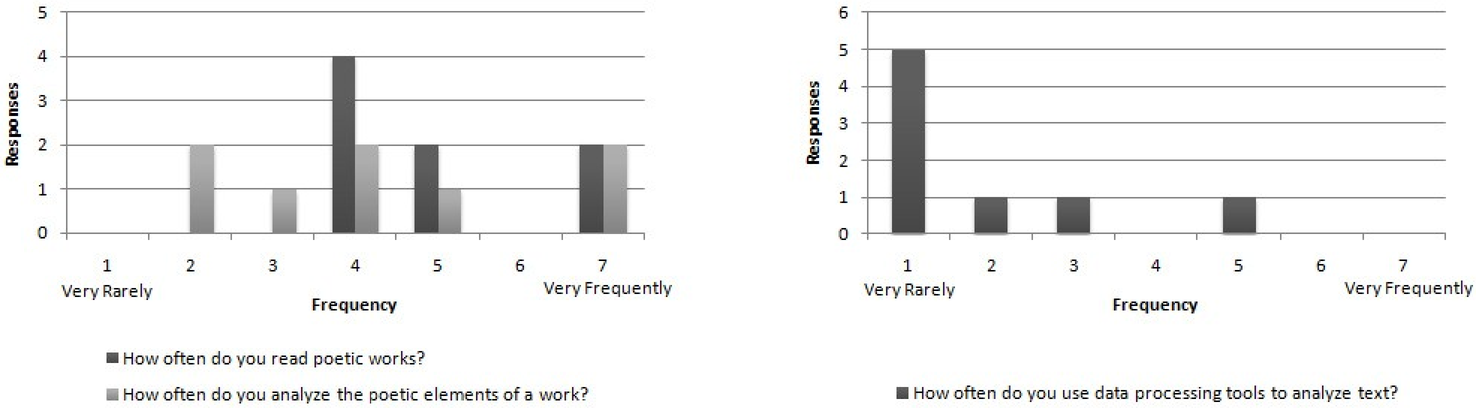

Fig. 7 . Survey Responses

There was confusion on the power of the search and filtering tools, as it was not clear if it was capable of searching for specific words or a patterns. Furthermore, we noted that some of the users felt the labels were not clear. Based on some user suggestions, we replaced some of the variable names with clearer ones. For example, the "content" label was originally "text", and the "wave" label was originally "tone".

Finally, we noted an even distribution of users willing to use this project in the future. Overall, the project was described as solid and intuitive.

XI. Future Work

The work presented here only represents the fundamental elements and initial ideas for visualizing spoken language features with VerseVis. While these contributions serve as proof of concept, there are many ideas for future work which build off the conceptual origins of the VerseVis solution. These plans can be categorized into several major groups.

A. *Rendering Options*

To improve the usability and intuitiveness of VerseVis tools we would like to improve the zoom interface by providing a slider instead of drop a down menu, add dragging functionality to panning, and in the overview picture we would like to add an outline which indicates the current viewing area.

VerseVis was initially designed to include a panel which displays text so that users could investigate details of the data without the need to zoom in. Several of the survey participants also mentioned a desire for comments, footnotes, and reference tools which we could incorporate into this panel.

In addition to the column and line rendering options currently provided by VerseVis, we would like to add another row option to the visualization. VerseVis currently expands the horizontal dimensions of the visualization to accommodate additional columns, but with many columns this results in a significant amount of white space when viewing all the documents. If we introduce a third division we could utilize this space. For example we might render documents by row, chapters by column, and sentences by line.

One of the limitations of VerseVis is that it considers all the details of a text, down to the phonemes of each word. When viewing very large bodies of text VerseVis may still need to render individual phonemes, which can have a major impact on the visualization performance and application responsiveness. In the future, we would like to develop abstracted rendering based on aggregated data at higher levels of the document hierarchy so that as viewers view large amounts of data, the visualization can balance the needs appropriately.

B. *Color Features and Additional Representation*

Straightforward additions like coloring column borders for each file and assigning them based on some statistical information relevant to its contents would allow users to easily identify column information. This would be particularly useful with many columns that are shown in non-linear order. Another simple addition would be to include custom palettes for color rendering of data. VerseVis colors for phoneme and words are currently generated automatically, but custom palettes would allow users to associate colors with data based on information not available to VerseVis. For large data ranges this might be impractical. Therefore it would be important to allow automatic colors by default and give users options to save palettes for future use.

We also envision utilizing hue, saturation, and value (HSV) components of color to visualize complex patterns in the text. The HSV color space would allow users to visualize additional dimensions of data from text at different levels of the document hierarchy.

One expert suggested that VerseVis provide an option for audio representation of data. This could be incorporated in a similar manner to the use of HSV color space, but instead using frequency and amplitude to represent information about the data.

C. *Filtering and Searching*

The VerseVis architecture is designed to incorporate complex searches and filters at multiple levels of the document hierarchy. We anticipate that these features would be a huge asset to the system.

First, VerseVis has modules that allow searching for a specific data query from various data fields at any level of the document hierarchy. In addition, we would like to add an interface to find patterns which could allow users to search for patterns among words, POS, stress, or any field of data. Finally, we would also like to support searching for groups of words. For example, a user might want to search and highlight a group of words consisting of delighted, ecstatic, elated, exalted, and other similar words in order to identify the happy words in a document. Search results could be highlighted or marked using a color chosen by the user, and multiple queries could be pinned to the visualization.

VerseVis also allows any level of the document hierarchy to be filtered based on some criteria. We would like to allow users to create multiple filters on the visualization with the option to also color the criteria or sort by the criteria. The criteria used for filtering could include frequency of specific query terms and patterns as well as the length of the data contained at the filtered level. This could be used to filter out phrases that are too long or too short, or it could let users sort columns by the number of lines.

XII. Conclusion

In this paper, we described VerseVis, a system that visualizes spoken features in poetry. Using data from Mary Text-to-Speech, we were able to provide users with the ability to identify interesting text patterns within literary corpora.

Feedback from usability test participants show that representing phonemes and accents through colors and height is a useful tool for getting an insight on a body of text. However, it also shows that there could be more work done on this project to increase utility and appeal.

Future work in this area include interfacing directly with Mary Text-to-Speech, providing an audio representation of the data, and improving the current system.

Acknowledgment

We would like to thank Dr. Tanya Clement and Dr. Ben Shneiderman for their mentorship, input, guidance, and infinite time and patience.

References

[1] Miller, N., Chung Wong, P., Brewster, M., Foote, H., 'TOPIC ISLANDS – A Wavelet-Based Text Visualization System'. 9th IEEE Visualization 1998 (VIS'98). Pages: 189 - 196, October 1998.

[2] Weber, W., 'Text Visualization - What Colors Tell About a Text', Proceedings of the 11th International Conference of Information Visualization, Pages: 354-362, July 2007

[3] Shiping Huang; Ward, M.O; Rudensteiner, E.A; 'Coordinated and Multiple Views in Exploratory Visualization', 2005. (CMV 2005). Proceedings. Third International Conference on Coordinated and Multiple Views in Exploratory Visualization. Pages 63-74.

[4] Don, A., Zheleva, E., Gregory, M., Tarkan, S., Auvil, L., Clement, T., Shneiderman, B., Plaisant, C., 'Discovering Interesting Usage Patterns in Text Collections: Integrating Text Mining with Visualization', ACM 16th Conference on Information and Knowledge Management. 2007.

[5] Fekete, Jean-Daniel; Dufournaud, Nicole. “Compus Visualization and Analysis of Structured Documents For Understanding Social Life in the 16th Century.”Proceedings of the fifth ACM Conference on Digital Libraries. 2000.

[6] Schneiderman, Ben. “The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization.” IEEE Conference on Visual Languages, 336-343. 1996.

[7] M. Schröder & J. Trouvain (2003). The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching. *International Journal of Speech Technology, 6, pp. 365-377.*

[8] Catherine Plaisant, James Rose, Bei Yu, Loretta Auvil, Matthew G. Kirschenbaum, Martha Nell Smith, Tanya Clement, and Greg Lord. 2006. Exploring erotics in Emily Dickinson's correspondence with text mining and visual interfaces. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries* (JCDL '06). ACM, New York, NY, USA, 141-150.

[9] Nitin Madnani. (2005). Emily: A Tool for Visual Poetry Analysis. http://www.umiacs.umd.edu/~nmadnani/emily/emily.pdf

[10] Ed H. Chi, Lichan Hong, Michelle Gumbrecht, Stuart K. Card. (2005). ScentHighlights: highlighting conceptually-related sentences during reading, Proceedings of the 10th international conference on Intelligent user interfaces, California, 272-274. http://www2.parc.com/istl/members/hong/publications/iui2005.pdf

[11] Keim, D.A.; Oelke, D.; , "Literature Fingerprinting: A New Method for Visual Literary Analysis," *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on* , vol., no., pp.115-122, Oct. 30 2007-Nov. 1 2007

XIII. Credits

Division of duties:

- Myers, Austin: contributed ideas and discussion, editing and writing of essay, exploring related work, coding, video
- Milton, Leslie: contributed ideas and discussion, editing and writing sections in the report, exploring related work, and setup the SVN repository.
- Lu, Christine: contributed ideas and discussion, writing and editing of essay, exploring related work, coding

Appendix A: Survey Questions

What degree are you seeking?

What is your focus of study?

How often do you read poetic works?

Very Rarely	1	2	3	4	5	6	7	Very Frequently

How often do you analyze the poetic elements of a work?

Very Rarely	1	2	3	4	5	6	7	Very Frequently

How often do you use data processing tools to analyze text?

Very Rarely	1	2	3	4	5	6	7	Very Frequently

Which features of VerseVis did you find most interesting?

Which features of VerseVis did you find the least interesting?

For future versions of VerseVis, how important would you consider each of the following features?

What additional features would you like to see included in VerseVis?

Would you use a tool like VerseVis?