Interactive Exploration of Versions across Multiple Documents

Chang-Han Jong, Prahalad Rajkumar, Behjat Siddiquie
chjong@umd.edu, prahalad@cs.umd.edu, behjat@cs.umd.edu

May 08, 2008

***Abstract*—Researchers in literature need to compare several versions of one or more poems or articles for investigating their historical and literary significance. Current tools do not adequately support their requirements. We address some of these issues by developing MultiVersioner, a tool designed to interactively analyze multiple documents, each consisting of several versions. Additionally, MultiVersioner enables users to search for entities such as words, phrases and lines, and facilitates the analysis of their frequency patterns. We have extensively used visualization principles such as color-coded highlighting and overview with details on demand. Promising feedback has been received from a domain expert. MultiVersioner also has potential applications in several other domains.**

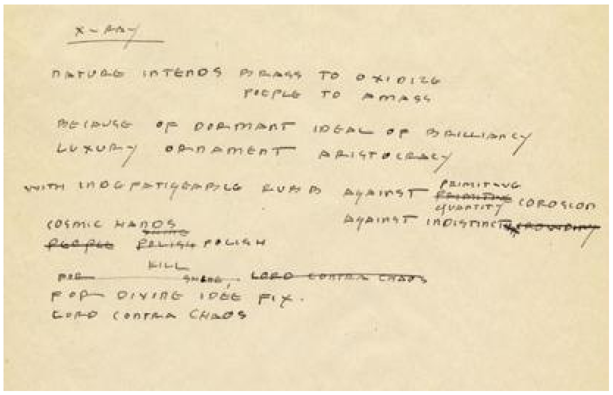
***Index Terms*—versions, comparison, multiple documents, literature, visualization**

I. INTRODUCTION

THE need to compare two or more documents arises in a variety of situations. Some instances include detection of plagiarism in academic settings, comparing versions of computer programs, and comparing the flow of history in the wiki setting such as wikipedia articles. Extensive research has been performed on comparing documents with each other based on their contents [7], [8], [10] and there also exist several tools such as windiff to visually compare a pair of documents. However, little work has been done on providing an effective visual interface to facilitate the comparison of more than two documents simultaneously. Versioning Machine (http://www.v-machine.org) by Schreibman et al. is a web-based interface that provides the facility to view multiple versions of a document, along with the changes across versions. Motivated by Versioning machine (VM), we build a tool MultiVersioner that facilitates viewing several documents at once, and provides the user with a rich set of relevant information regarding their comparison.

Our primary user is Tanya Clement, a graduate student in the English Department at the University of Maryland, who is also affiliated with the Maryland Institute of Technology in the Humanities (MITH). As part of her research Ms. Clement studies works by experimental poets. The works of these experimental writers consist of multiple unsequenced and cross-referenced versions that represent experiments in progress, rather than the final version. Ms. Clement analyzes how words change (appear, reappear, and disappear) not only across different versions of one poem, but across different versions of different poems to gain insights into the poet's experimental style. Figure 1 contains an example of a sample version of a poem showing additions and deletions. Ms. Clement uses VM to aid her in comparing different versions of a poem. The goal of this project was to develop an interface that built upon VM by providing additional features that facilitate such literary analysis .

The remainder of the report is organized as follows. Section 2 discusses other work related to our topic. Section 3 describes our program in detail, along with various aspects associated with it. Section 4 covers the qualitative evaluation of our program. We offer directions for future work and conclude our report in section 5.


X-RAY

NATURE INTENDS BRASS TO OXIDIZE
PEOPLE TO AMASS

BECAUSE OF DORMANT IDEAL OF BRILLIANCY
LUXURY ORNAMENT ARISTOCRACY

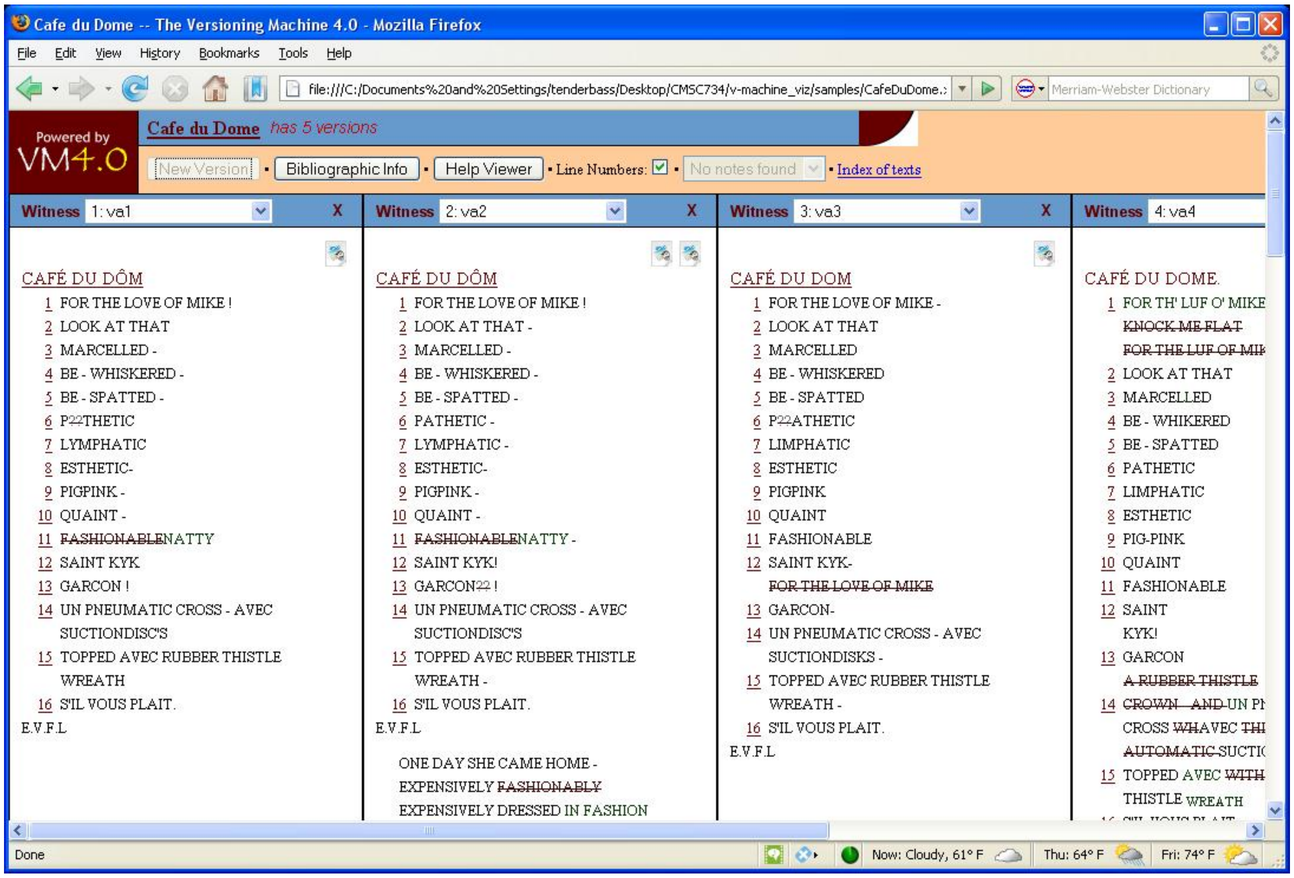
WITH INDEFATIGABLE RUB AGAINST PRIMITIVE ~~[illegible]~~ QUANTITY CORROSION
AGAINST INDISTINCT ~~[illegible]~~
COSMIC HANDS
~~PEOPLE~~ ~~RELISH~~ POLISH
KILL
~~FOR~~ ~~SHEEN~~ ~~LORD CONTRA CHAOS~~
FOR DIVINE IDEE FIX.
LORD CONTRA CHAOS

Figure 1. Sample Version of a Poem

II. RELATED WORK

Viégas et al. [2] address the issue of monitoring history changes in wikipedia articles. Their work overlaps to some extent with our project and provided us with some insight on representing change in documents. Also relevant is the discussion of movement of text within a document, particularly when content is added or deleted from certain versions of a document. [4] covers the aspects of movement in text extensively. An important aspect of our tool was highlighting the similarities and differences across several documents. Very little literature was available on this topic, but we were able to borrow a few pointers from *ScentHighlights* by Chi et al. [5]. ScentHighlights is a tool developed to highlight words based on searches performed by the user. This idea of using highlighting to display search results was incorporated into MultiVersioner.

Plagiarism detection and source code comparison are areas that have been extensively researched, we reviewed plenty of


Cafe du Dome -- The Versioning Machine 4.0 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/Documents%20and%20Settings/tenderbass/Desktop/CMSC734/v-machine_viz/samples/CafeDuDome.
Merriam-Webster Dictionary
Powered by VM4.0
Cafe du Dome has 5 versions
New Version
Bibliographic Info
Help Viewer
Line Numbers: ✓
No notes found
Index of texts
Done
Now: Cloudy, 61° F
Thu: 64° F
Fri: 74° F

Witness 1: va1

CAFÉ DU DÔM
1 FOR THE LOVE OF MIKE !
2 LOOK AT THAT
3 MARCELLED -
4 BE - WHISKERED -
5 BE - SPATTED -
6 P~~22~~THETIC
7 LYMPHATIC
8 ESTHETIC-
9 PIGPINK -
10 QUAINT -
11 ~~FASHIONABLE~~NATTY
12 SAINT KYK
13 GARCON !
14 UN PNEUMATIC CROSS - AVEC SUCTIONDISC'S
15 TOPPED AVEC RUBBER THISTLE WREATH
16 S'IL VOUS PLAIT.
E.V.F.L

Witness 2: va2

CAFÉ DU DÔM
1 FOR THE LOVE OF MIKE !
2 LOOK AT THAT -
3 MARCELLED -
4 BE - WHISKERED -
5 BE - SPATTED -
6 PATHETIC -
7 LYMPHATIC -
8 ESTHETIC-
9 PIGPINK -
10 QUAINT -
11 ~~FASHIONABLE~~NATTY -
12 SAINT KYK!
13 GARCON~~22~~ !
14 UN PNEUMATIC CROSS - AVEC SUCTIONDISC'S
15 TOPPED AVEC RUBBER THISTLE WREATH -
16 S'IL VOUS PLAIT.
E.V.F.L

ONE DAY SHE CAME HOME -
EXPENSIVELY ~~FASHIONABLY~~
EXPENSIVELY DRESSED IN FASHION

Witness 3: va3

CAFÉ DU DOM
1 FOR THE LOVE OF MIKE -
2 LOOK AT THAT
3 MARCELLED
4 BE - WHISKERED
5 BE - SPATTED
6 P~~22~~ATHETIC
7 LIMPHATIC
8 ESTHETIC
9 PIGPINK
10 QUAINT
11 FASHIONABLE
12 SAINT KYK-
~~FOR THE LOVE OF MIKE~~
13 GARCON-
14 UN PNEUMATIC CROSS - AVEC SUCTIONDISKS -
15 TOPPED AVEC RUBBER THISTLE WREATH -
16 S'IL VOUS PLAIT.
E.V.F.L

Witness 4: va4

CAFÉ DU DOME.
1 FOR TH' LUF O' MIKE
~~KNOCK ME FLAT~~
~~FOR THE LUF OF MIK~~
2 LOOK AT THAT
3 MARCELLED
4 BE - WHIKERED
5 BE - SPATTED
6 PATHETIC
7 LIMPHATIC
8 ESTHETIC
9 PIG-PINK
10 QUAINT
11 FASHIONABLE
12 SAINT
KYK!
13 GARCON
~~A RUBBER THISTLE~~
14 ~~CROWN AND~~ UN PN
CROSS ~~WH~~AVEC ~~THI~~
~~AUTOMATIC~~ SUCTI
15 TOPPED AVEC ~~WITH~~
THISTLE WREATH

Figure 2. Versioning Machine 4.0

literature and tools. CHECK [7], MOSS [14] and Copyscape [15] are some plagiarism detection tools. *CHECK* [7], which concentrated on the algorithmic approaches to document comparison, is representative of the work in the area of plagiarism detection. Brin et al. [8] describe copy detection mechanisms in digital content. The Visual Code Navigator by Lommerse et al. [10] is a recent work that focuses on source code investigation.

BasketLens [11], *FeatureLens* [12], and *Emily* [13] are all projects developed at the University of Maryland that are closely related to our project. The goal of all the three projects is to facilitate exploration of text-based documents. Emily [13] is a tool designed to visualize and analyze unstructured human-generated text like poetry, in particular poetry by Emily Dickinson. BasketLens, which is based on Emily, explored the eroticism in Emily Dickinson's poem. BasketLens [11] introduced the concept of baskets, defined as "a group of words unified by some concept". An example of a basket could be the category *flowers.* A search performed on this basket would return instances of *rose*, *daisy* etc. *FeatureLens* [13] was a class project for Information Visualization in Spring 2006, that facilitated pattern finding in text collections by providing visualizations of the results of text mining algorithms.

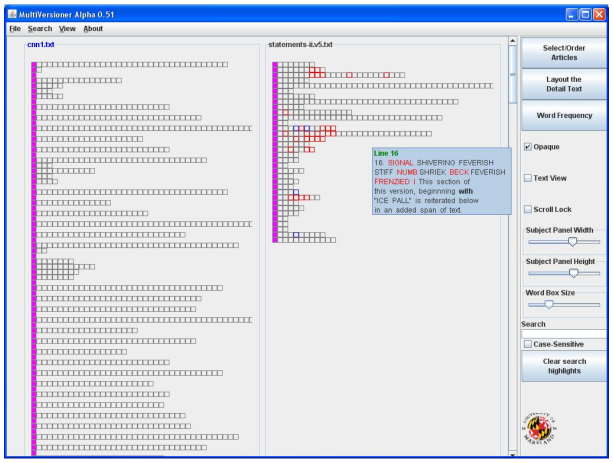
The following sources covering eclectic topics were also of interest to us. Hongyuan Zha and Xiang Ji [9] adopt a novel approach to compare multilingual documents, by making use of bipartite graphs. Veksler and Gray [6] try to predict where the human eye is likely to catch information, and provide suggestions to make use of their predictions in order to aid viewing of information.

A. *Versioning Machine*

As mentioned earlier, Versioning Machine developed by Schreibman et al. is the tool that motivated our work. The paper [1] contains a detailed description of the mechanics of VM and how the tool facilitates comparing different versions of a document. VM requires all versions of a poem to be encoded in a single XML document. The XML document encodes the various additions and deletions made across different versions. VM enables the display of several versions of a document on the screen (Figure 2), and also denotes the additions and deletions in one version with respect to the other versions. It also provides a text based overview showing some meta information of the poem such number of versions, name of each version and the name of poet.

One of the primary requirements of Ms. Clement was to compare not only versions of a single poem, but also multiple poems. VM can display the versions of just one document at a time. To open another document, all versions of the current document have to be closed first. Another requirement of Ms. Clement was to search for specific words and analyze their patterns. VM does not provide any search capabilities, both across versions as well as across documents. MultiVersioner provides an interface to facilitate the visualization of several versions of multiple documents simultaneously. Additionally, it also has search and statistical analysis capabilities to facilitate literary analysis.

III. Description of the Interface



Line 16
16. SIGNAL SHIVERING FEVERISH STIFF NUMB SHRIEK BECK FEVERISH FRENZIED I This section of this version, beginnning **with** "ICE PALL" is reiterated below in an added span of text.

Figure 4. Viewing two long versions

A. *Background*

The goal of our project is two-fold, to provide an effective overview of the content and size of all documents, as well as to provide a detailed display, along with a variety of search capabilities. This is in accordance with the Shneiderman Mantra *Overview first, zoom and filter, details on demand* [16].

MultiVersioner is implemented in Java 6.0 using the Swing GUI toolkit. It contains a built-in parser to parse the poem versions that are encoded as XML documents. Loading an XML document opens all the versions of a poem in separate *version panels* and multiple such poems can be opened simultaneously. Additionally, a text file containing a single version can be directly loaded. Version panels are displayed in the central part of the interface with a tool panel located on the right. The name of the version appears on top of the respective version panel. The names of all the versions of a particular document are displayed in the same color in order to group them. Users can switch between a box view and a text view.

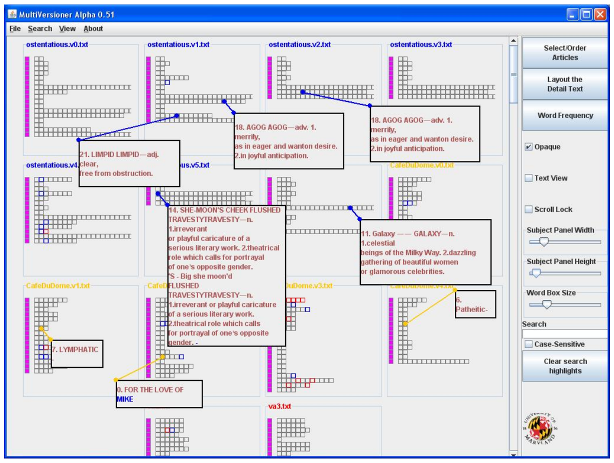
B. *Overview (box view)*

In the overview, words are denoted by equal sized boxes (Figure 3). Mousing over a box pops up a tooltip containing the entire sentence, with the current word being shown in bold (Figure 4). In the tooltip, words added in the current version are shown in blue, and words deleted are shown in red.

Clicking on a box brings up a *detail window* (Figure 6) containing the entire sentence. The purpose of the detail window is to display a sentence of interest on the screen, analogous to a post-it note. Users have the option of making the detail windows either opaque or transparent by toggling the *Opaque* checkbox. The button *Layout the Detail Text* aligns all the detail windows together. A line is drawn between a detail window and its corresponding location in the version panel, to keep track of its origin. A detail window could be closed either by right clicking on the detail window and choosing the *close* option, or by dragging the detail window away from the screen boundary, causing it to go out of the screen.

C. *Text View*

Using word boxes to represent words is used primarily to obtain an overview of all the documents. To explore the versions in detail, a representation displaying the actual sentences, instead of word boxes, is preferred (Figure 5). Switching to the text representation can be done by checking the *Text View* checkbox. In the next two sections, we describe features such as word search, line search and frequency table which are common to both the box view and the text view.



21. LIMPID LIMPID—adj. clear, free from obstruction.

18. AGOG AGOG—adv. 1. merrily, as in eager and wanton desire. 2.in joyful anticipation.

18. AGOG AGOG—adv. 1. merrily, as in eager and wanton desire. 2.in joyful anticipation.

14. SHE-MOON'S CHEEK FLUSHED TRAVESTYTRAVESTY—n. 1.irreverant or playful caricature of a serious literary work. 2.theatrical role which calls for portrayal of one's opposite gender. 'S - Big she moon'd FLUSHED TRAVESTYTRAVESTY—n. 1.irreverant or playful caricature of a serious literary work. 2.theatrical role which calls for portrayal of one's opposite gender. -

11. Galaxy —— GALAXY—n. 1.celestial beings of the Milky Way. 2.dazzling gathering of beautiful women or glamorous celebrities.

6. Patheitic-

7. LYMPHATIC

0. FOR THE LOVE OF MIKE

Figure 6. Pop-up Detail Window

D. *Search*

MultiVersioner provides extensive search features to facilitate comparison between several documents (Figure 7). The very basic search feature is the word search. A search bar is provided where the user can type in a word or a phrase to be searched across all documents. A checkbox is provided to make a search case-sensitive. An alternative way to perform a word search is to right click the word (word box in the case of box view) to be searched. Search results are color-coded, and the instances of a searched word in all documents are highlighted using the same color. Contrasting colors are used for different searches to distinguish between search results. A search history is available at the bottom of the panel in the right. The facility to clear a particular search result as well as all the search results across the version panels is provided.

A line search feature is available as well. An anchor box is present at the beginning of each line. Right clicking it triggers a line search, where the specified line will be searched across all documents. Our line search algorithm, which works by computing a similarity score between a pair of lines, and returns all the lines that are reasonably similar to the specified line. Similarity between a pair of lines is obtained by first computing the number of words common to the two lines,

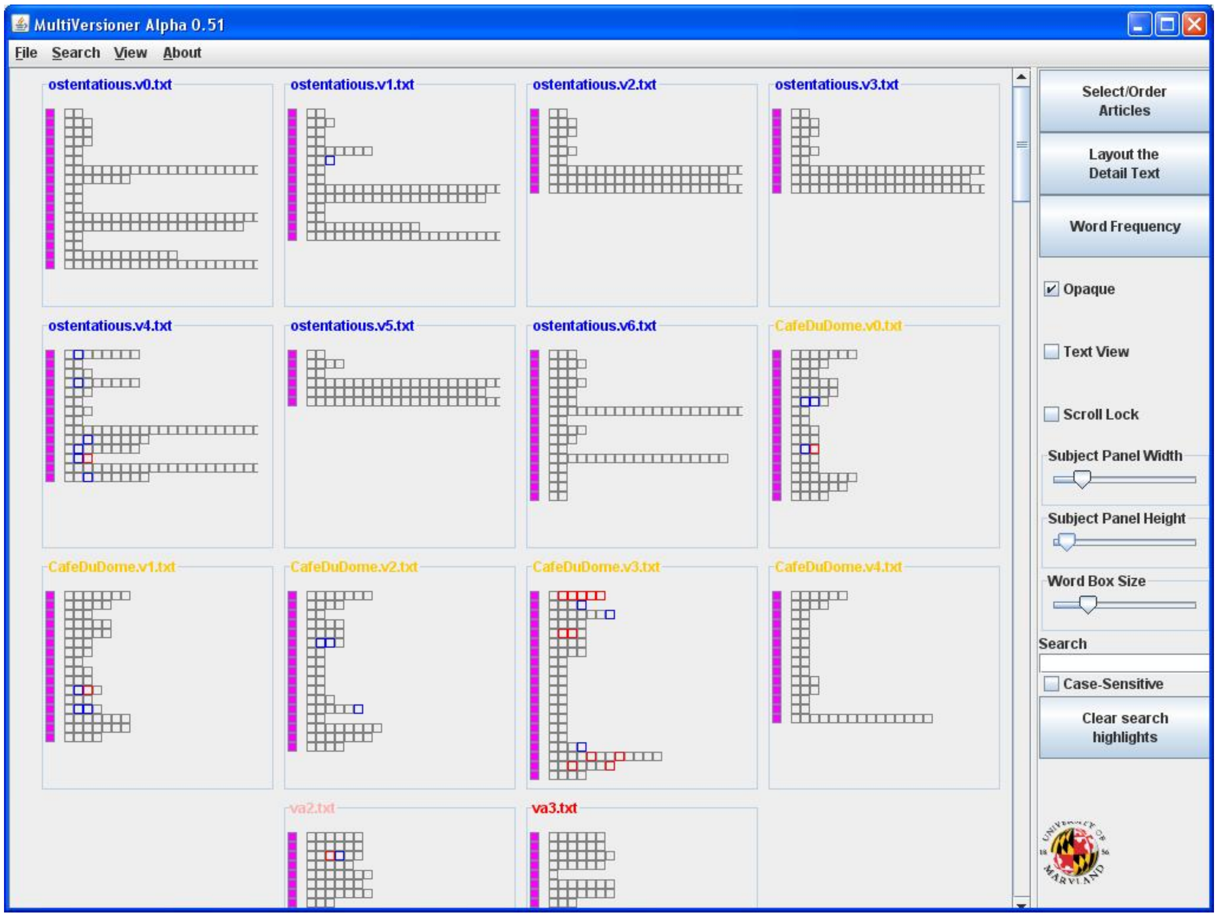
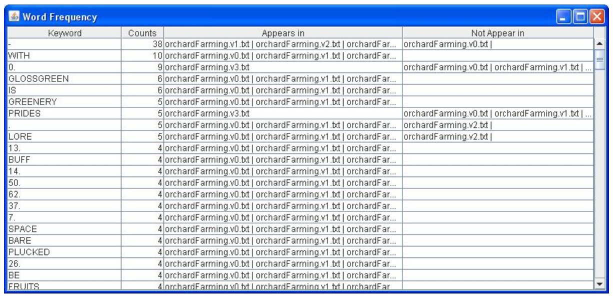


Figure 3. Overview of many versions and documents in MultiVersioner

and then normalizing this by the sum of the number of words in both the lines. Two lines l_1 and l_2 are said to be similar if similarity$(l_1, l_2) > \Theta$, we used $\Theta = 0.5$. Words of length ≤ 2 were excluded from this computation. This simple algorithm proved quite effective. Several other algorithms were tried out, but due to the arbitrary nature of the changes between the versions, they did not work well. As with the word search, the line search colors matching lines throughout all the documents using the same color.

E. Word Frequency Table


Word Frequency

Keyword	Counts	Appears in	Not Appear in
-	38	orchardFarming.v1.txt | orchardFarming.v2.txt | orchardFar...	orchardFarming.v0.txt |
WITH	10	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
0.	9	orchardFarming.v3.txt	orchardFarming.v0.txt | orchardFarming.v1.txt | ...
GLOSSGREEN	6	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
IS	6	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
GREENERY	5	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
PRIDES	5	orchardFarming.v3.txt	orchardFarming.v0.txt | orchardFarming.v1.txt | ...
.	5	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	orchardFarming.v2.txt |
LORE	5	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	orchardFarming.v2.txt |
13.	4	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
BUFF	4	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
14.	4	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
50.	4	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
62.	4	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
37.	4	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
7.	4	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
SPACE	4	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
BARE	4	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
PLUCKED	4	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
26.	4	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
BE	4	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	
FRUITS	4	orchardFarming.v0.txt | orchardFarming.v1.txt | orchardFar...	

Figure 8. Frequency Table

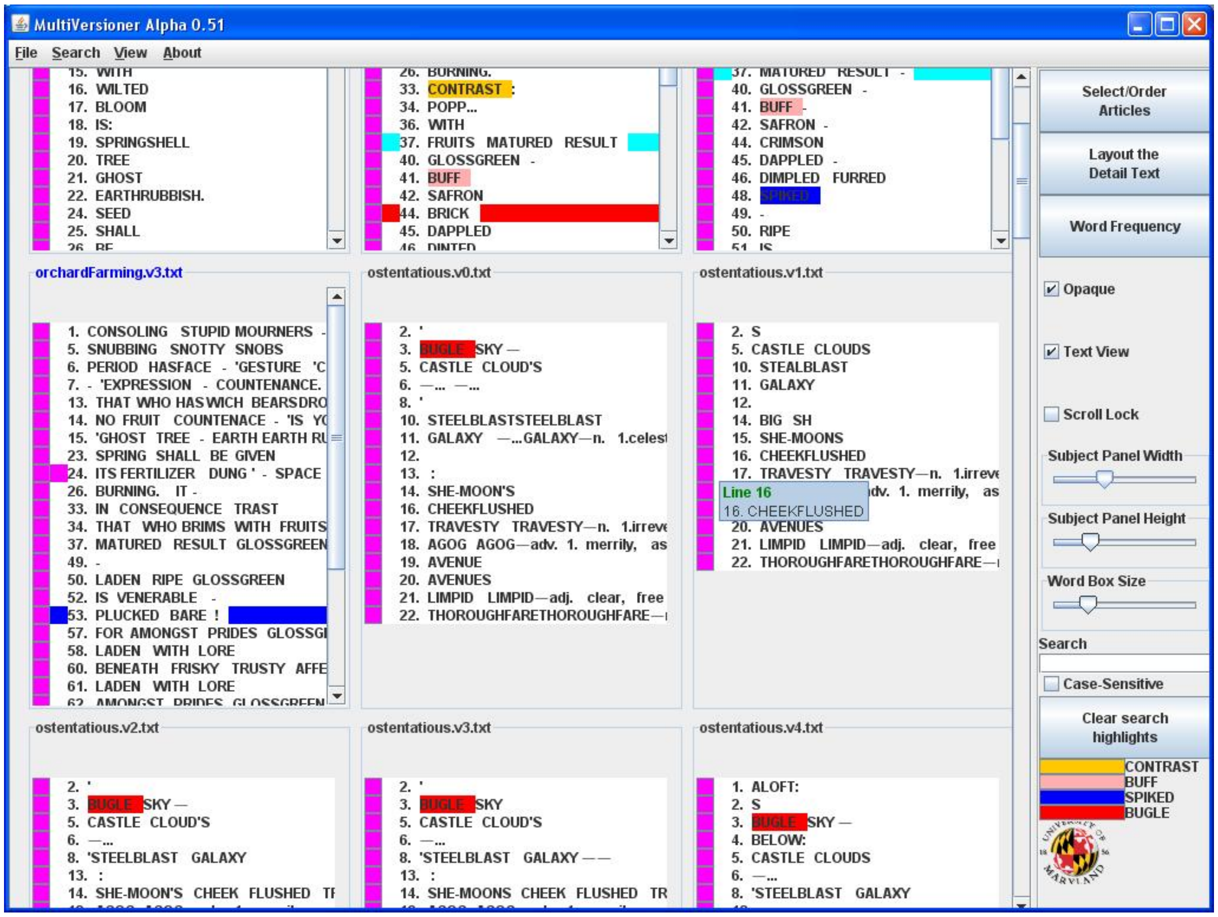
MultiVersioner computes a frequency table (Figure 8) containing the number of occurrences of each unique word in all documents and their versions . When comparing different versions of a document or comparing different documents that are related, researchers in literature need to identify unique and common words and sentences. We believe that an approach as simple as a frequency table listing is powerful in providing insight. Given the number of articles, and the number of times that a word appears in each article, users can know which words are common across documents and which ones are unique to a single document.

F. Other features

This section wraps up the discussion of the program interface by mentioning the various other features available in MultiVersioner. There are sliders available to control the version panel height, width and the sizes of the word boxes. We also have a scroll lock that becomes functional if the any of the version panels contain long documents. Choosing the scroll lock by checking the *Scroll Lock* checkbox synchronizes the scrollable documents with each other. If one document is scrolled, other documents scroll as well. A brief user manual is available under the *About* menu outlining the functionalities of all the controls of MultiVersioner.

G. An Example Scenario

Here is an example of the proceedings of a typical MultiVersioner session.



```
15. WITH
16. WILTED
17. BLOOM
18. IS:
19. SPRINGSHELL
20. TREE
21. GHOST
22. EARTHRUBBISH.
24. SEED
25. SHALL
```

```
26. BURNING.
33. CONTRAST :
34. POPP...
36. WITH
37. FRUITS  MATURED  RESULT
40. GLOSSGREEN  -
41. BUFF
42. SAFRON
44. BRICK
45. DAPPLED
```

```
37. MATURED  RESULT  -
40. GLOSSGREEN  -
41. BUFF  -
42. SAFRON  -
44. CRIMSON
45. DAPPLED  -
46. DIMPLED  FURRED
48. SPIKED
49. -
50. RIPE
```

orchardFarming.v3.txt

```
1. CONSOLING  STUPID MOURNERS  -
5. SNUBBING  SNOTTY  SNOBS
6. PERIOD  HASFACE  -  'GESTURE  'C
7. -  'EXPRESSION  -  COUNTENANCE.
13. THAT WHO HAS WICH  BEARSDRO
14. NO FRUIT  COUNTENACE  -  'IS  YC
15. 'GHOST  TREE  -  EARTH EARTH RU
23. SPRING  SHALL  BE  GIVEN
24. ITS FERTILIZER  DUNG '  -  SPACE
26. BURNING.  IT -
33. IN  CONSEQUENCE  TRAST
34. THAT  WHO BRIMS  WITH  FRUITS
37. MATURED  RESULT  GLOSSGREEN
49. -
50. LADEN  RIPE  GLOSSGREEN
52. IS VENERABLE  -
53. PLUCKED  BARE  !
57. FOR AMONGST  PRIDES  GLOSSGI
58. LADEN  WITH  LORE
60. BENEATH  FRISKY  TRUSTY  AFFE
61. LADEN  WITH  LORE
```

ostentatious.v0.txt

```
2. '
3. BUGLE  SKY —
5. CASTLE  CLOUD'S
6. —...  —...
8. '
10. STEELBLASTSTEELBLAST
11. GALAXY  —...GALAXY—n.  1.celest
12.
13.  :
14. SHE-MOON'S
16. CHEEKFLUSHED
17. TRAVESTY  TRAVESTY—n.  1.irreve
18. AGOG  AGOG—adv.  1. merrily,  as
19. AVENUE
20. AVENUES
21. LIMPID  LIMPID—adj.  clear,  free
22. THOROUGHFARETHOROUGHFARE—
```

ostentatious.v1.txt

```
2. S
5. CASTLE  CLOUDS
10. STEALBLAST
11. GALAXY
12.
14. BIG  SH
15. SHE-MOONS
16. CHEEKFLUSHED
17. TRAVESTY  TRAVESTY—n.  1.irreve
dv.  1. merrily,  as
20. AVENUES
21. LIMPID  LIMPID—adj.  clear,  free
22. THOROUGHFARETHOROUGHFARE—
```

ostentatious.v2.txt

```
2. '
3. BUGLE  SKY —
5. CASTLE  CLOUD'S
6. —...
8. 'STEELBLAST  GALAXY
13.  :
14. SHE-MOON'S  CHEEK  FLUSHED  TF
```

ostentatious.v3.txt

```
2. '
3. BUGLE  SKY
5. CASTLE  CLOUD'S
6. —...
8. 'STEELBLAST  GALAXY ——
13.  :
14. SHE-MOONS  CHEEK  FLUSHED  TR
```

ostentatious.v4.txt

```
1. ALOFT:
2. S
3. BUGLE  SKY —
4. BELOW:
5. CASTLE  CLOUDS
6. —...
8. 'STEELBLAST  GALAXY
```

Figure 5. Text View

1) The user, let us call her Sarah, wishes to compare and explore some poems. Sarah uses the mouse throughout the interaction. She opens an XML file that contains encoded versions of the poem *Autumn*. MultiVersioner parses through the XML file, and displays the four versions of *Autumn* on the screen.
2) By default, MultiVersioner displays the overview of the versions, i.e. with a word box denoting a word. Depending on the size of the articles and the screen, Sarah has the option of using the *Subject Panel Width* and *Subject panel Height* sliders to adjust the width of the articles.
3) Sarah then decides to search for the word *world*. She types *world* in the search box and hits Enter. The word box for each instance of *world* in each document turns blue. Then she searches for the adjacent word *withers* and each instance of *withers* turns red.
4) Sarah wants to search for the line *The sun burns out*. She does so by right clicking on its anchor box, which is a magenta colored box present at the beginning of the line. Each sentence similar to the selected sentence will be highlighted cyan.
5) At this stage, Sarah finds the line *The sun burns out* to be of interest. She opens up a detail window by clicking on the line anchor and displays the line on the screen. A line is drawn from the detail window to its originating word box, to relate the two entities together.
6) After Sarah has finished performing a fair amount of exploration and has opened a number of detail windows on the screen, she has the option of choosing the *Layout the detail texts* button to automatically arrange all the detail windows.
7) Sarah decides that she does not need a particular detail window, and removes it by dragging it out of the screen boundary.
8) Sarah wants to move from the overview to the text view by checking the *Text View* check box.
9) Sarah opens another poem *Nocturne*, the versions of which are encoded in an XML file as well. The three versions of *Nocturne* are displayed on the screen.
10) She searches for the word *time*, which performs a search across the four versions of *Autumn*, as well as the three versions of *Nocturne,* coloring the instances pink. In a similar manner, Sarah continues to explore through these two poems.

IV. EVALUATION

The main aim of the evaluation was to improve our tool by enhancing its utility for our target users. The evaluation was a continuous process during which the input received from participants was incorporated into the tool and further feedback was obtained. This process was repeated several

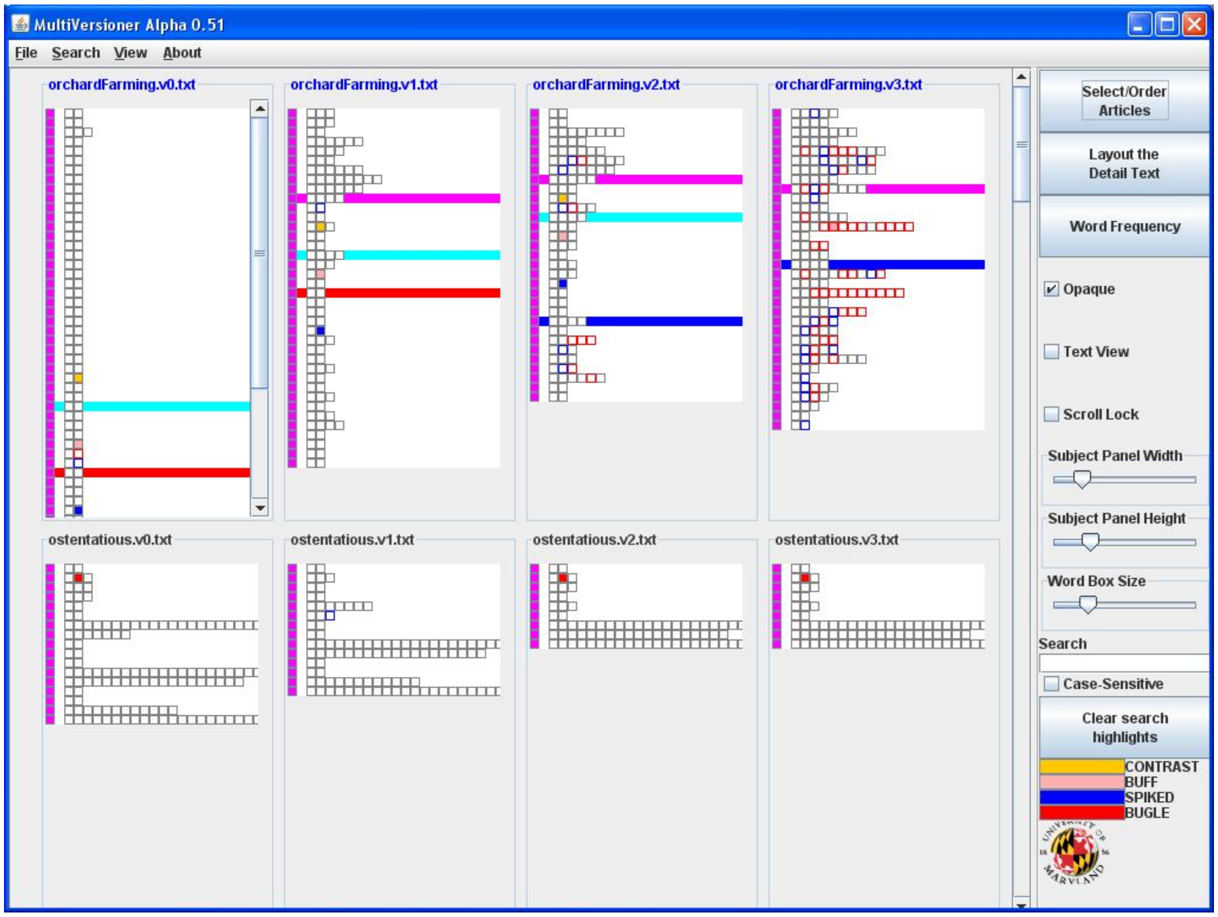


Figure 7. Highlighting word and line search results

times. We received inputs from two types of users: 1) A Domain expert whom the tool was aimed at and 2) Domain novices.

A. Domain Expert

As mentioned earlier, Tanya Clement was our domain expert. We started by understanding her requirements and followed up with several evaluation sessions. During these sessions, the protocol followed was first performing a demonstration of the new features implemented, followed by usability testing and feedback regarding the advantages and disadvantages of MultiVersioner with respect to the VM. Since we were working with only one user, the nature of our evaluation was qualitative.

1) Document layout: Ms. Clement wanted all versions of a single document to be readily distinguishable from other documents. After brainstorming several possibilities with her, we decided that coloring the titles of all the version panels associated with the same document with the same color was appropriate. She also wanted to have the ability to move the version panels around by dragging them, and placing them at a location of her convenience as this would enable her to place two versions of interest together and make observations.

2) Search: The ability to search for words across the documents was greatly appreciated by Ms. Clement. She also provided positive feedback on the color-coded highlighting of the search results. However, she seemed a bit confused by the line search results as our algorithm did not classify certain results as matches.

3) Frequency Table and Statistical Analysis: One of the features she required was displaying the unique words in each version. To incorporate this, we construct a frequency table which enables users to view the frequency of occurrence of words across different document versions. From the frequency table, words unique to a particular version are identified and highlighted. In the next stage she suggested that a user be allowed to see the spatial occurrences of a particular word by clicking it in the frequency table. The frequency table forms the basis for developing several statistical analysis and data mining techniques to aid the analysis.

4) Text View vs Box View: Ms. Clement stressed that she prefers seeing the actual words, rather than they being represented as word boxes. If that is the case, then she said she does not have to think back and forth, attempting to associate boxes with words. She added that boxes are helpful for a high-level view consisting of a large number of documents and versions, but is was more useful for her analysis.

5) Miscellaneous features: Among all the other features, she found the synchronized scrolling helpful. To identify the origin of detail windows we implemented a feature where double clicking a detail window causes the originating version panel to flicker. She found this to be distracting. Ms. Clement suggested that we link the detail window to its originating location in the version panel by drawing a line. On implementation, this turned out to be an effective solution. Ms. Clement was amused by the feature of making the detail window disappear by drag-and-dropping it out of the screen.

B. Feedback from others

We also received suggestions from people who were not experts in English literature and poetry. The evaluation method primarily consisted of a demonstration, with the participants occasionally playing with the controls and then telling us what they liked or disliked. Most of the inputs we received were specific to visualization aspects such as layout and color schemes.

During the entire course of this project we got regular feedback from Dr. Shneiderman. Initially our main focus was on providing an overview by using equal-sized word boxes to denote the words of a sentence. Dr. Shneiderman suggested that users across most domains would prefer the actual text being displayed, rather than seeing the word boxes. In our final version, we were able to capture the flavor of Dr. Shneiderman's suggestion. His other suggestions included the use of a gray background, rather than a black background, which would distract the user's attention from other aspects of the interface.

We also performed a live demonstration before our fellow CMSC 734 colleagues, all of whom have a reasonable knowledge of several advanced visualization concepts, and received additional feedback from them. They liked the idea of searching for words across versions and synchronized scrolling across document versions. Feedback included associating the detailed windows with their position in the documents, using less contrasting colors for the document background and allowing users to see the words instead of boxes. One suggestion was the effective utilization of the entire screen space to display the documents. This prompted us to implement sliders enabling the users to manually control the height and width of the documents. We also offer additional suggestions for optimizing the screen space utilization.

There were a couple of suggestions which were offered by both domain experts as well as others. The first suggestion was the preference of seeing actual text in the version panels instead of the word boxes. They reported that correlating the word boxes to the poems themselves is confusing on occasions. As we received this comment from several sources, we appreciated the importance of displaying the actual text. Also, we had initially failed to relate the detail windows with the word boxes they originated from; both user groups pointed out that we needed to have a mechanism to associate the detail windows with their originating word boxes.

V. CONCLUSIONS AND FUTURE WORK

There exist several tools for comparing documents, articles and software code based on some similarity metric and a few of these enable the user to perform statistical analysis on the documents [11][12]. A related area that has not been extensively researched is facilitating the comparison of multiple versions of a document. VM is probably one of the first tools that enables this. We built upon VM in the following two ways:

- Allowing the user to compare multiple documents, each of which consists of multiple versions
- Providing the ability to search for entities such as words and lines across the documents and versions and analyze the frequency patterns of these entities

In the process, we also explored visualization techniques and gained some important insights. We studied the use of color to perform effective highlighting and for discovering patterns. The aspect of providing an overview, with details available on demand, was implemented, which helped us appreciate some of the issues involved. We have demonstrated the effectiveness of our tool for comparing poems, but the underlying concepts are applicable to several other domains such as comparison of patents, discovering the evolution of laws over time and end user license agreements (EULAs). The search and comparison methods can be suitably modified to discover meaningful relationships between the respective entities.

As MultiVersioner was developed as a part of a semester project, we were able to implement only a limited number of features. We now outline the various possibilities that can be explored in the context of visualizing multiple document versions. As our primary domain of focus was poems, MultiVersioner works best for data of small sizes. Future work could involve addressing the problem of visualizing long documents, which could for example be used in the contexts of comparing different versions of programs or different versions of wikipedia articles. The possibility of utilizing the entire screen space to fit all open documents should be examined as well. In this case, opening new documents or closing existing ones, should result in the program dynamically resizing the documents into equal segments that fit the screen. While doing this one would also like to restrict changes in configuration of the layout and ensure that the users do not lose context. Other areas that can be studied are using color for grouping and highlighting hierarchical entities like documents, paragraphs, lines and words. Adding additional features inspired by [11], [12] and adapting them for the purpose of comparing document versions to aid the statistical analysis would substantially increase the utility of MultiVersioner. There is ample potential to research effective interfaces in this area, and we explored some of the possibilities here by implementing MultiVersioner.

REFERENCES

[1] Susan Schreibman, Amit Kumar, Jarom McDonald. (2003). The Versioning Machine, *Literary and Linguistic Computing,* 18(1), 101-107

[2] Fernanda B. Viégas, Martin Wattenberg, Kushal Dave. (2004). Studying cooperation and conflict between authors with history flow visualizations, *Proceedings of the SIGCHI conference on Human factors in computing systems*, Vienna, Austria, 575-582.

[3] Nancy E. Miller, Pak Chung Wong, Mary Brewster, Harlan Foote. (1998). TOPIC ISLANDS TM – A Wavelet-Based Text Visualization System, *Proceedings on the conference of visualization*, North Carolina, 189-196.

[4] Kenton O'Hara, Abigail Sellen. (1997). A Comparison of Reading Paper and On-Line Documents, *Proceedings of the SIGCHI conference on Human factors in computing systems*, Georgia, 335-342.

[5] 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.

[6] Vladislav Daniel Veksler, Wayne D. Gray. (2007). Mapping semantic relevancy of information displays, *Proceedings of ACM CHI 2007 Conference on Human Factors in Computing Systems* v.2 2729-2734

[7] Antonio Si, Hong Va Leong, Rynson W. H. Lau. (1997). CHECK: a document plagiarism detection system, *Proceedings of the 1997 ACM symposium on Applied computing*, California, 70-77.

[8] Sergey Brin, James Davis, Hector Garcia-Molina. (1995). Copy detection mechanisms for digital documents, *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, California, 398-409.

[9] Hongyuan Zha, Xiang Ji. (2002). Correlating multilingual documents via bipartite graph modeling, *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, Finland.

[10] Gerard Lommerse, Freek Nossin, Lucian Voinea, Alexandru Telea. (2005). The Visual Code Navigator: An Interactive Toolset for Source Code Investigation, *in Proc. IEEE InfoVis'05, IEEE CS Press*, 24-31.

[11] Darya Filippova. (2007). BasketLens: interface for document visualization and exploration, *Independent study conducted with Ben Shneiderman and Catherine Plaisant.*

[12] Anthony Don, Elena Zheleva, Machon Gregory, Sureyya Tarkan, Loretta Auvil, Tanya Clement, Ben Shneiderman, Catherine Plaisant. (2007). Discovering interesting usage patterns in text collections: integrating text mining with visualization, *HCIL Technical report.*

[13] Nitin Madnani. (2005). Emily: A Tool for Visual Poetry Analysis.

[14] MOSS, http://theory.stanford.edu/~aiken/moss, retrieved 04-10-2008

[15] Copyscape, http://www.copyscape.com, retrieved 04-10-2008

[16] Ben Shneiderman. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization. *IEEE Conference on Visual Languages*, 336-343.