

Interactive Exploration of Versions across Multiple Documents

Chang-Han Jong

University of Maryland
chjong@umd.edu

Prahalad Rajkumar

University of Maryland
prahalad@cs.umd.edu

Behjat Siddique

University of Maryland
behjat@cs.umd.edu

Tanya Clement

University of Maryland
tclement@umd.edu

Catherine Plaisant

University of Maryland
plaisant@cs.umd.edu

Ben Shneiderman

University of Maryland
ben@cs.umd.edu

Introduction

The need to compare two or more documents arises in a variety of situations. Some instances include detection of plagiarism in academic settings and comparing versions of computer programs. Extensive research has been performed on comparing documents based on their content (Si et al., 1997; Brin et al., 1995) 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. *The Versioning Machine* (Schreibman et al., 2003) is a web-based interface that provides the facility to view multiple versions of a document, along with the changes across versions. Motivated by the Versioning Machine (VM), we build a tool *MultiVersioner* that facilitates viewing multiple versions of multiple documents at once, and provides the user with a rich set of information regarding their comparison. The primary user during the development of *MultiVersioner* was Tanya Clement, a

doctoral candidate in English at the University of Maryland, who researches the works of experimental poets.

Related Work

ScentHighlights (Chi et al., 2005) has demonstrated the effectiveness of using color-coded highlighting to display the similarities and differences across documents. There exist literature (Brin et al., 1995) and tools like *CHECK* (Si et al., 1997) and *MOSS* (MOSS) on plagiarism and source code comparison, which are relevant to our work. *FeatureLens* (Don et al., 2007) facilitates pattern finding in text collections by providing visualizations of the results of text mining algorithms.

In Tanya Clement's research, she compares not only versions of a single poem, but also multiple versions across several 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. VM also does not provide any search capabilities.

Description of the Interface

Background

The two-fold goal of *MultiVersioner* is 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, in accordance with the *Info-Viz Mantra Overview first, zoom and filter, details on demand* (Shneiderman 1996).

MultiVersioner is implemented in Java 6.0 using the Swing GUI toolkit. It uses the same input format file as VM, an XML file, containing information about the various additions and deletions made across all versions of a document. *MultiVersioner* contains a built-in parser to parse these XML files. Loading an XML file opens all the versions of a poem in separate *version panels* and multiple such documents can be loaded simultaneously. 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 together.

Overview

In the overview, words are denoted by equal sized boxes. Hovering over a box pops up a tooltip containing the entire sentence, with the current word being displayed in bold. 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 1) containing the entire sentence. The purpose of the de-

tail window is to display a sentence of interest on the screen, analogous to a post-it note. The detail windows can be made either opaque or transparent, and can either be moved around, or aligned 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 it and choosing close, or by simply dragging it out of the screen.

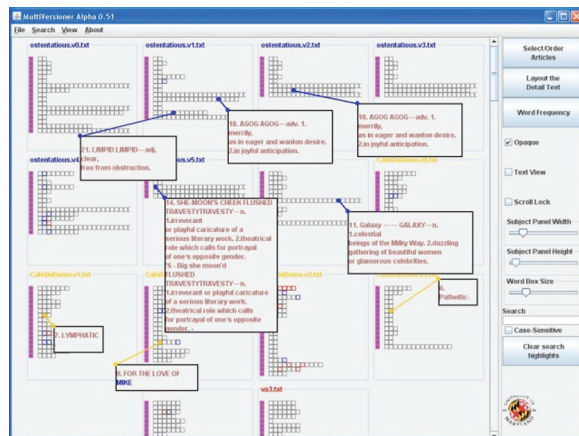


Fig. 1 Overview of multiple versions of two poems. Sentences of interest are shown using detail windows, which are linked to their respective version panels by lines.

Text View

Word boxes are 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 2).

Search

The basic search feature is the word search (Figure 2). A search bar is provided where the user can type in a word or a phrase to be searched across all documents. A search can be made case-sensitive if desired. Alternatively, a word can be searched by right-clicking an instance of it. Inspired by ScentHighlights (Chi 2005 et al., 2005), Search results are color-coded. The instances of a searched word in all documents are highlighted using the same color. A search history as well as the facility to clear search results is available.

A line search feature is available as well. The similarity of a pair of lines is computed by taking into account the number and relative positions of the words common to them. Right-clicking the anchor-box present at the beginning of a line triggers a line search and lines similar to the specified line across all documents are highlighted.

Word Frequency Table

MultiVersioner computes a frequency table 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. It has been shown that an approach as simple as a frequency table listing is powerful in providing insight by letting users know which words are common across documents and which ones are unique to a single document (Filipova, 2007).

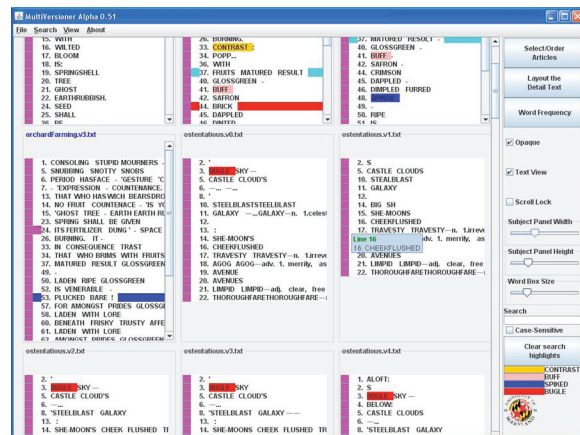


Fig. 2 A text view of versions of two poems. Four different searches for the words “contrast”, “buff”, “spiked” and “bugle” are performed across all versions of both poems. Each instance of searched word is highlighted using the same color in all documents.

Other features

There are sliders available to control the version panel height, width and the sizes of the word boxes. MultiVersioner also has a scroll lock, used when the documents are long, to synchronize the scrolling of the documents with each other.

User Feedback and Future Work

The first three authors of the abstract were the developers (students in an Information Visualization class). The last three authors used the prototypes several times during the development and provided suggestions. Examples are given below.

Users reported that they could not easily distinguish versions of a single document from versions of other documents. We achieved this by using the same color for the titles of all the version panels associated with a the same document. The ability to search for words across the documents was greatly appreciated and the color-coded highlighting of the results was found helpful, but overall

they commented that softer colors would be more pleasing. Users liked the synchronized scrolling between windows. Still it was difficult to associate the detail window to its originating location, so some visual linking was suggested (e.g. drawing a line or matching color highlighting). Regarding the choice between Text View and Overview, users stressed that it is preferable to see the actual words rather than seeing the abstract word boxes. Still, they acknowledged the benefit of the overview when dealing with a large number of documents and versions. The overview was more helpful for a high-level view and orientation, the text view was more useful for analysis.

In summary we believe that our prototype illustrates that it is possible to facilitate visual comparison. We built on the Versioning Machine by allowing users to compare multiple documents, each of which consisting of multiple versions. We also provide the ability to search for entities such as words and lines across the documents and versions and analyze their frequency patterns. MultiVersioner was designed to compare small poems, and future work needs to address the problem of longer documents. Utilizing the entire screen space, by dynamically resizing all open documents to fit the screen, might be a promising direction. Further information about this project including a more detailed report, slides, a demo video and software can be found at https://wiki.cs.umd.edu/cmssc734_08/index.php?title=MultiVersioner.

References

- Brin, S., Davis J., and Garcia-Molina H. (1995). *Copy detection mechanisms for digital documents*, Proc. of the 1995 ACM SIGMOD international conference on Management of data, 398-409.
- Chi, E. H., Hong, L., Gumbrecht, M., and Card S. K. (2005). *ScentHighlights: highlighting conceptually-related sentences during reading*, Proc. of the 10th International Conference on Intelligent User Interfaces, 272-274.
- Don A., Zheleva, E., Gregory, M., Tarkan, S., Auvil, L., Clement, T., Shneiderman, B., and Plaisant, C. (2007). *Discovering interesting usage patterns in text collections: integrating text mining with visualization*, Proc. of the sixteenth ACM Conference on Information and Knowledge Management, 213-222.
- Filippova, D. (2007). *BasketLens: interface for document visualization and exploration*, <http://www.cs.umd.edu/hcil/textvis/basketlens/>.

MOSS. <http://theory.stanford.edu/~aiken/moss>, re-

trieved 04-10-2008

- Schreibman, S., Kumar, A., and McDonald, J. (2003). "The Versioning Machine." *Literary and Linguistic Computing*, 18(1), 101-107 (<http://www.v-machine.org>)
- Shneiderman, B. (1996). "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization." *IEEE Conference on Visual Languages*, 336-343.
- Si, A., Leong, H. V., Lau, R. W. H. (1997). CHECK: a document plagiarism detection system, *Proc. of the 1997 ACM symposium on Applied computing*, 70-77.