DOI: 10.1111/j.1467-8659.2011.01923.x

Eurographics / IEEE Symposium on Visualization 2011 (EuroVis 2011)
H. Hauser, H. Pfister, and J. J. van Wijk
(Guest Editors)

Volume 30 (2011), Number 3

Semantic-Preserving Word Clouds by Seam Carving

Yingcai Wu[1], Thomas Provan[1], Furu Wei[2], Shixia Liu[2], and Kwan-Liu Ma[1]

[1]The Visualization and Interface Design Innovation (VIDi) Research Group, University of California, Davis
[2]Mircrosoft Research Asia, Beijing, China

Abstract

Word clouds are proliferating on the Internet and have received much attention in visual analytics. Although word clouds can help users understand the major content of a document collection quickly, their ability to visually compare documents is limited. This paper introduces a new method to create semantic-preserving word clouds by leveraging tailored seam carving, a well-established content-aware image resizing operator. The method can optimize a word cloud layout by removing a left-to-right or top-to-bottom seam iteratively and gracefully from the layout. Each seam is a connected path of low energy regions determined by a Gaussian-based energy function. With seam carving, we can pack the word cloud compactly and effectively, while preserving its overall semantic structure. Furthermore, we design a set of interactive visualization techniques for the created word clouds to facilitate visual text analysis and comparison. Case studies are conducted to demonstrate the effectiveness and usefulness of our techniques.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques

1. Introduction

Text data is produced at an incredible rate because of the progress in computing power and storage capacity over the last decades. Regardless of the domain, business professionals and scholars often need to understand and analyze document collections to facilitate their decision making. One particularly interesting area is to visually analyze and compare multiple related documents across a text corpus. The need to visually illustrate multiple related documents at once arises naturally in many real-world applications. For example, Twitter posts are a useful resource for researching public opinion on companies and competitors. A business consultant may want to compare public opinion on the major products of several companies such as Microsoft, Google, IBM, and HP, and then include the comparison results into his business report. A market analyst on the other hand may want to find out the major advantages and disadvantages of a newly released product/service. To help users gain insights into related documents, it is important to allow them to visually examine and compare the documents interactively.

Word clouds are an effective means for users to understand the major content of a document instantly. However, traditional word clouds are not suitable for comparative document visualization, as they often present words in random or alphabetical order. Although they are proliferating in visualization and text analysis, it is difficult for them to tell how the words related to a certain topic vary from those related to other topics, since these words scatter in the word clouds. Users would have to visually search for their desired words in the word clouds for comparison, thus introducing additional overhead. However, creating semantic-preserving word clouds is challenging. We have attempted to create context-preserving word clouds with a force-directed algorithm [CWL*10], but this approach has several drawbacks. First, the resulting word clouds may not be stable; a slight change of the input words may result in very different word clouds. Second, the created word clouds may have very irregular shapes. In comparative visualization, word clouds with regular shapes, such as rectangles, are preferred by users. Finally, the original semantic relations among words may be destroyed in the resulting word clouds, which may confuse or mislead users in visualization.

To address these issues, we introduce a new semantic-preserving word cloud generation method based on tailored seam carving, a well-established content-aware image resizing technique. In this work, we define a topic as a group of semantically similar keywords in a text corpus. Two keywords are considered as semantically similar if they co-occur

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

in a corpus. The goal of our work is to create a compact word cloud while preserving the keyword semantic relations by putting the semantically similar keywords close to one another. Our method first uses an energy function to find low energy regions in the layout, then it iteratively carves out a left-to-right or top-to-bottom seam to remove empty space between words. The major feature of this method is that it can generate compact word clouds while preserving the original semantic relations among words. To further facilitate visual comparison of the related documents, we use *Bubble Sets* [CPC09] and spreadsheet-style visualization.

With this work, we contribute to the ongoing research on text visualization by visually illustrating the similarities and differences among multiple groups of documents with the help of semantic-preserving word clouds. Consequently, we design a new semantic-preserving word cloud generation algorithm using an adapted seam carving technique. Another contribution is that we use bubble sets and spreadsheet visualization to allow users to analyze and compare different documents from different levels of details.

2. Related Work

Existing text visualization can be generally classified into three categories: word-oriented, document-oriented, and theme-oriented methods. The word-oriented methods such as Wordle [VWF09] mainly use word clouds to visually summarize document collections. However, they cannot show the semantic relationships between words or the temporal patterns. To tackle this problem, ManiWordle [KLKS10] provides flexible control on the layout result of Wordle, which allows the user to manipulate the overall layout, as well as the layout of the individual words. Another system, SparkClouds [LRKC10], integrates sparklines into a word cloud to convey the temporal patterns between multiple word clouds. Collins et al. [CVW09] presented a special visual representation called parallel tag clouds to visualize differences amongst facets of large text corpora. Strobelt et al. [SOR*09] designed a new compact visualization for summarizing the key semantics of a document by a mixture of images and important key terms. Our previous work [CWL*10] couples a trend chart with word clouds to illustrate temporal content evolution patterns.

The document-oriented approaches [Ren94, SGL08, OST*10, CSL*10] focus on transforming a collection of text documents into a visual illustration. FacetAtlas designed by Cao et al. [CSL*10] allows users to visually analyze and explore the document with multiple dimensions in rich text corpora. Oesterling et al. [OST*10] introduced a two-stage method for topological analysis of a document collection.

The theme-oriented methods automatically derive a set of themes from a document collection [PG10], then visualize the analytic results. ThemeRiver [HHWN02] and its derivatives [LZP*09, DGWC10, SWL*10] use a river metaphor to depict the thematic variation over time within a text corpus. Rose et al. [RBC*09] developed a flow-style story visualization to help users track stories over time. Fisher et al. [FHRH08] used a simple yet effective line chart to show both the changes in the concepts and the relationships between the concepts.

These general text visualization methods only allow the user to compare different documents at the word, document, or theme level. In contrast, our work enables the user to reason about the similarity and differences of document collections from multiple perspectives (e.g., content, and relationships among different topics in a word cloud).

3. System Overview

Figure 1 shows our visualization system. It has three major components: a preprocessing component, a word cloud layout component, and a visualization component. The preprocessing component first extracts all keywords from a collection of documents. Then the similarity value between any two extracted keywords is calculated, and a dissimilarity matrix is built accordingly. After that, a 2D word scatterplot is created by multidimensional scaling based on the dissimilarity matrix. Finally, it places keywords on the word scatterplot, and uses a force-directed algorithm to eliminate the word overlaps to create a preliminary word cloud layout.

The word cloud component is responsible for creating a compact word cloud from the preliminary layout. To create a word cloud for a document, the component first removes irrelevant keywords that do not appear in the document from the preliminary layout. This usually results in a sparse word cloud with much whitespace. Our system uses tailored seam carving [AS07] to pack the sparse word cloud while preserving the relative positions of important keywords, thus creating a semantic-preserving word cloud for the document.

The visualization component applies Bubble Sets [CPC09] to the packed word clouds to enhance the semantic relationships. It can also present the word clouds to users in a spreadsheet-style layout, which general users are familiar with. Thus, it easily enables side-by-side visual document comparison. In addition, we design some user interactions for the special word cloud spreadsheet. These interactions allow users to interactively search, compare, merge, and split the word clouds inside the spreadsheet.

4. Preprocessing

In preprocessing, our system extracts important keywords from an input document collection, creates a 2D word scatterplot, and generates a preliminary word layout.

4.1. Keyword Extraction

We extract keywords from a document collection using a graph based algorithm called LexRank [ER04]. Specifically,

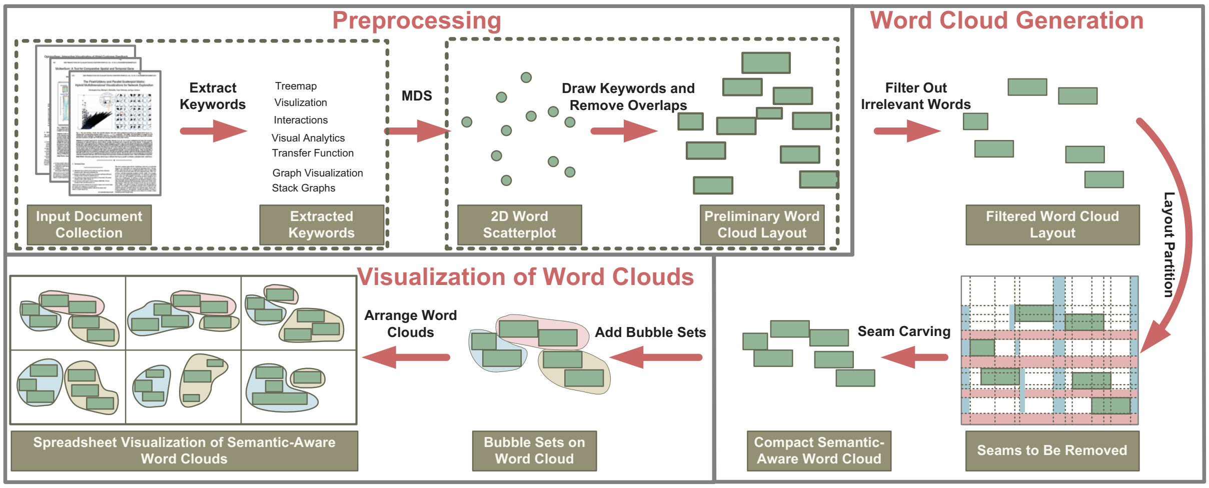

Preprocessing
Word Cloud Generation
Extract Keywords
Treemap
Vizulization
Interactions
Visual Analytics
Transfer Function
Graph Visualization
Stack Graphs
MDS
Draw Keywords and Remove Overlaps
Filter Out Irrelevant Words
Input Document Collection
Extracted Keywords
2D Word Scatterplot
Preliminary Word Cloud Layout
Filtered Word Cloud Layout
Layout Partition
Visualization of Word Clouds
Arrange Word Clouds
Add Bubble Sets
Seam Carving
Spreadsheet Visualization of Semantic-Aware Word Clouds
Bubble Sets on Word Cloud
Compact Semantic-Aware Word Cloud
Seams to Be Removed

Figure 1: *System overview: the system has three major components: a preprocessing component for extracting keywords and creating a preliminary layout, a word cloud generation component for creating compact and semantic-preserving word clouds, and a visualization component for visualizing the word clouds.*

we first employ an open source toolkit [Ope] to split each document into a set of sentences which are then tokenized into a collection of words. Each word is further stemmed using the Porter Stemmer [Por80]. Next, we build a word co-occurrence graph $G = (V,E)$ by taking each word as a node in V and adding an edge $e(i, j)$ into E if words i and j appear in the same sentence. The weight of $e(i, j)$ is aggregated by the number of co-occurrences of these two words, i and j. We then run the LexRank algorithm on G to get the stationary distribution of v in the Markov Chain defined by G. We also run noun phrase chunking to detect the phrases using the Stanford parser [Sta], and then we use the phrases together with other unit words as nodes in the LexRank algorithm. Let $R_{1\times|V|}$ be the ranking vector, then R is defined as,

$$\begin{aligned} R &= d\cdot M\bullet R+(1-d)\cdot\vec{p}, P\bullet R=\lambda\cdot R \\ P &= d\cdot M+(1-d)\cdot\vec{p}\bullet 1^T \end{aligned} \tag{1}$$

- d is the damping factor and set to 0.85 according to [ER04].
- $M_{|V|\times|V|}$ is the normalized word co-occurrence matrix where the summation of each column equals 1.
- $\vec{p}$ is a probability vector and set to $[\frac{1}{|V|}]_{1\times|V|}$.
- R is the eigenvector corresponding to the largest eigenvalues (i.e. $\lambda = 1$ here) of M, and it can be computed by the power iteration method as used in [ER04].

The words with high rank values in R are the keywords.

4.2. 2D Word Scatterplot

We use a method proposed by Schutze [Sch98] to measure the similarity between two words with the occurrence matrix M used in Equation (1). Let M_i and M_j indicate rows i and j of M; the similarity between the corresponding word i and j can be computed by the cosine measure. With the similarity measure, we can build a dissimilarity matrix for all the extracted keywords, which can then be used by multidimensional scaling to create a 2D scatterplot for conveying the semantic relations between keywords.

4.3. Preliminary Word Layout

If we draw keywords with different font sizes on the 2D word scatterplot, we may have a cluttered word cloud with words overlapping one another. Thus, we design a simplified force-directed model adapted from [CWL*10] to remove the word overlaps by exerting only a repulsive force for any two overlapping words. The model starts with Delaunay Triangulation which creates a triangle mesh from the initial word layout. The algorithm then applies a repulsive force for any two overlapping words. Finally, it creates an adjusted word layout where the word overlaps are removed and the semantic relations are largely preserved. The created word layout is a preliminary and sparse layout for showing the overall semantic relations among all the extracted keywords.

5. Semantic-Preserving Word Cloud Generation

This section describes our seam carving method for creating compact and semantic-preserving word clouds.

5.1. Word Cloud Generation by Seam Carving

A preliminary word cloud layout is semantically meaningful, since words that often appear together in the documents

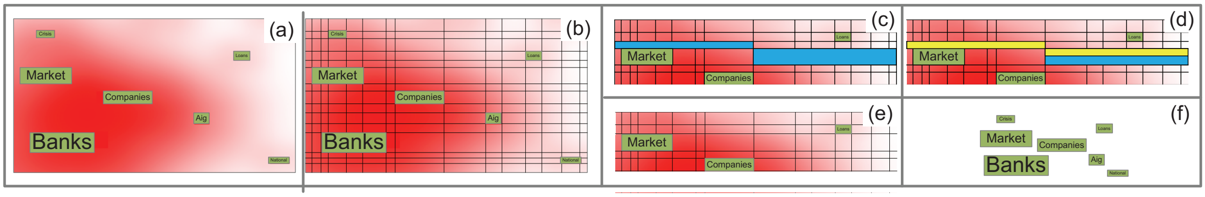


Figure 2: *Illustration for Seam Carving: (a) a sparse word cloud layout with a Gaussian importance field; (b) layout partitioned by the bounding boxes of the words; (c) an optimal seam (marked in blue), a connected path of zones, from left to right is selected; (d) seam pruning to obtain a seam with an identical width (yellow seam); (e) the word cloud layout after removing the yellow seam in (d); (f) the resulting compact and semantic-preserving word cloud after the seam carving optimization.*

are arranged nearby in the layout. Although this can facilitate visual comparison of multiple word clouds, there is often much whitespace left among words, thus wasting the screen space and limiting its capability for comparing multiple word clouds side by side.

Inspired by seam carving [AS07] , we designed an algorithm for removing the whitespace while maintaining the original semantic structure of word clouds. Seam carving is a content-aware image resizing operator that can reduce or enlarge an image. It first estimates the importance of pixels by an energy function. Seam carving then iteratively selects a seam of low energy pixels crossing the image from top to bottom, or from left to right. By carving-out or inserting the selected seams in both directions, seam carving can change the size of the image successfully while preserving the image structure. Our algorithm works in a manner similar to content-aware image resizing by seam carving.

Energy Function Arbitrarily removing white space in the preliminary layout would change the semantic relations between words. An ideal approach to compressing sparse word clouds is to remove whitespace among words in a judicious manner, such that the related spatial positions among words are preserved. However, it is usually impossible to remove the empty regions without any change in the semantic structure of the word clouds. When an empty region is removed in a word cloud, other parts of the layout should be moved to cover the removed region, which often results in the change of spatial relations between the words. In other words, there is a tradeoff between the compactness and the semantic change. Intuitively, one solution to the problem is to minimize the semantic relation change of those more important words and sacrifice the less important words.

Thus, we designed an optimization algorithm to strike a balance between the compactness and the semantic change. The algorithm has two major parts: an energy function and an operator to optimize the tradeoff. The energy function is used to ensure the semantic relations of more important words are largely unaffected and tell the operator which part of the whitespace should be removed. The whitespace surrounded by more important words should have higher energy, while those surrounded by less important words should have lower energy. Therefore, we use a Gaussian distribution to imitate the influence of each word on its neighboring empty regions. The energy of a pixel is the accumulated gaussian value of all Gaussian distributions. Formally, the energy at pixel (x,y) can be defined as follows.

$$E(x,y) = \sum_{i=1}^{n} w_i \frac{1}{2\pi\sigma^2} e^{-((x-\mu_{x_i})^2)+(y-\mu_{y_i})^2)/2\sigma^2} \qquad (2)$$

where μ_{x_i} and μ_{y_i} are the positions of word i, $\sigma = 1$, and w_i is the normalized size of word i ranging from 0 to 1. The empty regions near important words with larger font size have higher energy than others. Figure 2(a) shows an energy field estimated by Equation (2).

Seam Carving Operator The energy function defined in (2) can help us find out which regions should be removed. A question then arises: how can we remove the low energy regions to minimize the change to the semantic structure? A simple solution that removes the low energy regions in ascending order in a word cloud does not work, as this would likely change the spatial relations among words dramatically. Thus, we need a judicious method to remove the whitespace. As we mentioned earlier, seam carving is a content-aware image resizing operator that we could use to pack word clouds. However, simply applying the original seam carving technique to pack word clouds would damage word integrity, as the seams may cross over the words. Additionally, the original seam carving algorithm is inefficient, especially for the preliminary sparse word layouts which are usually very large, since it carves out seams of one pixel width one by one, thus making it inappropriate for interactive visualization. We hereby design a new seam carving technique specifically for word clouds. It repeatedly removes a connected path of low energy zones rather than pixels to accelerate the performance. To preserve the word integrity, we let the connected path have an identical width and prevent the seams from passing through the words.

We start the algorithm by using the edges of the bounding boxes of all the words to partition the word cloud into a set of zones. This creates an $n \times m$ rectilinear grid of zones (see Figure 2 (b)). We then remove a seam of zones repeatedly to pack the word cloud. Formally, a seam is defined as a connected path of low energy empty zones from top to bottom (*vertical seam*), or left to right (*horizontal seam*). Let $Z = \{z(i,j) | 1 \le i \le n, 1 \le j \le m\}$ be the zone set and $z(i,j)$ represent the zone at the i^{th} row and the j^{th} column. We can formally define a horizontal seam and a vertical as:

$$s^x = \{z(x(j),j)\}_{j=1}^{m}, \text{s.t. } \forall j, |x(j)-x(j-1)| \le 1 \quad (3)$$

$$s^y = \{z(i,y(i))\}_{i=1}^{n}, \text{s.t. } \forall i, |y(i)-y(i-1)| \le 1 \quad (4)$$

where $x(j)$ is a function which maps an input column index j to a typical row index i, while $y(i)$ is a function which maps an input row index i to a typical column index j. Thus, a horizontal (or vertical) seam is an 8-connected path of zones from left to right (or top to bottom) with only one zone in each column (or row) of the layout.

With the energy function defined in (2), we can estimate the energy of each zone $E(z(i,j))$ by accumulating the energy of its internal pixels. Therefore, our goal is to seek the optimal s^* that minimizes this seam cost:

$$s^* = \min E(s) = \begin{cases} \min \sum_{j=1}^{m} E(z(x(j),j))) & \text{if } s = s^x; \\ \min \sum_{i=1}^{n} E(z(i,y(i)))) & \text{if } s = s^y. \end{cases}$$

Our algorithm finds the optimal vertical seam by dynamic programming. It computes the cumulative minimum energy E_c for all possible connected seams for each zone (i,j):

$$E_c(i,j) = E(i,j) + C \quad (5)$$

$$C = \min(E_c(i-1,j-1), E_c(i-1,j), E_c(i-1,j+1)) \quad (6)$$

Finally, we can find out the end of the optimal vertical seam from the minimum cumulative value in the last row. We can then backtrack from this minimum value on E_c to identify the path of the optimal seam. Finding the optimal horizontal seam is similar by dynamic programming. Figure 2 (c) shows an example where an optimal seam (marked in blue) is selected by dynamic programming.

To preserve word integrity, the algorithm should not select any zone that contains words. That is, only empty zones should be considered in the dynamic programming process. Assume the zone $z(x(j),j)$ has the minimum height in s^x. We prune the seam using the minimum height, such that its zones have identical widths (see the pruned seam in yellow in Figure 2 (d) for an example). This can also prevent the words being distorted when we remove the optimal seam. Figure 2 (e) is the layout where the pruned seam is removed.

Seam Carving Optimization The order of removing vertical and horizontal seams plays an important role in achieving an optimal packing. Different ordering strategies (horizontal seams first, vertical seams first, or alternating between them) often result in distinct packing results. To create an optimal packing, we transform the seam carving ordering problem into an optimization problem with an objective function:

$$\min \sum_{i=1}^{k} E(\alpha_i s_i^x + (1-\alpha_i) s_i^y) \quad (7)$$

where k is the number of seams to be removed, and $\alpha_i \in \{0,1\}$. If we remove a horizontal seam at step i, we let $\alpha_i = 1$, otherwise $\alpha_i = 0$. We again use dynamic programming to find out an optimal solution to the problem. Let $M(p,q)$ denote the minimum energy cost of removing p horizontal seams, and q vertical seams. Obviously, we have $M(0,0) = 0$. We can obtain a recursive function for dynamic programming as

$$M(p,q) = \min(M(p-1,q) + E(s^x(W_{n-p-1 \times m-q})), \\ M(p,q-1) + E(s^y(W_{n-p \times m-q-1}))) \quad (8)$$

where $W_{n-p-1\times m-q}$ and $W_{n-p\times m-q-1}$ represent two word clouds with an $n-p-1 \times m-q$ rectilinear and an $n-p \times m-q-1$ rectilinear grid of zones, respectively. $E(s^x(W))$ and $E(s^y(W))$ are the costs for removing the optimal horizontal seam and vertical seam, respectively.

We set $p = n-1$ and $q = m-1$ at the beginning of dynamic programming for packing a word cloud $W_{n\times m}$ with an $n \times m$ rectilinear grid of zones. As we mentioned earlier, our seam carving optimization has a special restriction, i.e., any seam should not cross over words in the layout. As a result, the recursive process of dynamic programming can stop at a certain step $M(r,c)$, when there is no appropriate seam that consists of only empty zones available for both $E(s^y)$ and $E(s^x)$. This leads to an optimal size $(r \times c\ )$ for packing the word cloud. Finally, we backtrack from $M(r,c)$ to $M(0,0)$ and remove the corresponding seams recorded in the previous process. Figure 2 (f) is a resulting semantic-preserving and compact word cloud generated by our algorithm.

Time Performance Analysis Our algorithm is more efficient than the original seam carving algorithm, because it removes a seam of zones rather than a seam of pixels every time. If a sparse word layout contains k keywords, we have an $n \times m$ rectilinear grid of zones where $n = m = 2k+1$. We need a running time of $O(mn) = O(k^2)$ to find an optimal seam from the layout. The total running time would be $O(k^2(r+c))$ where r and c denote the numbers of horizontal seams and vertical seams to be removed.

6. Visualization with Semantic-Preserving Word Clouds

This section describes a set of word cloud visualization techniques to facilitate comparative visualization of documents.

6.1. Bubble Set Visualization

A major advantage of word clouds over stack graphs for users to visually track topic evolution is that word clouds can reveal the semantic correlations among different topics. Our algorithm can maintain the semantic relations among words largely, but there is still a need to visually reveal the group relations in the word cloud. This is because different groups of keywords look much closer than before in resulting word clouds, making it difficult for users to visually distinguish between individual groups. Furthermore, some keywords of a group may be separated from the remaining keywords of the same group and surrounded by the keywords of other groups. Users may find this even more difficult to differentiate between word groups.

We employ a visualization technique called *Bubble Sets* proposed by Collins et al. [CPC09] for intuitively revealing multiple group relations in a word cloud. Bubble Sets uses a continuous isocontour to delineate set membership in a layout, and avoids adjusting the layout to improve set cluster continuity and density. This technique is suitable for our semantic-preserving word clouds, as we want to preserve the semantic relations among words without disrupting the primary layout. The groups (or clusters) of a word cloud are determined by clustering words in the corresponding preliminary 2D word scatterplot with k-means clustering. Bubble Sets can then be created over the word cloud using the group information as well as the individual word node information.

6.2. Spreadsheet-Style Visualization

Our word clouds are especially useful for comparing documents at the same time, as users can visually examine how a group of similar keywords that may be related to a certain topic changes in different documents. This leads to a spreadsheet-style visualization of word clouds for comparative visualization. In the spreadsheet, each row represents a document collection, while each column represents a subset of the documents. The spreadsheet enables users to compare multiple document collections interactively and visually. It supports the conventional spreadsheet user interactions such as search, sorting, and selection. In addition, we design two special user interactions, *merge* and *split*, to allow users to explore document collections at different scales.

- **Merge** Users can select multiple columns or rows of word clouds, and merge them. The system will first filter out the irrelevant keywords which do not appear in the selected word clouds from the preliminary layout, and create a new compact word cloud from the filtered preliminary layout by seam carving. Figure 3 (a)-(c) shows an example of merging two word clouds (Figure 3 (a)-(b)) for a new word cloud (Figure 3(c)).
- **Split** Users can select a column, and then choose multiple keyword groups on a word cloud of the column. The spreadsheet allows users to split the selected row by just dragging the selected keyword groups out of the word cloud. A new column will be added for holding the new word clouds of the selected keywords. Figure 3(d)-(f) presents an example of how we split a word cloud (Figure 3(d)) for two new word clouds (Figure 3(e) and (f)).

7. Experiments and Case Studies

We have implemented the tailored seam carving algorithm using Java and built our spreadsheet-style visualization system based on Prefuse. We tested our algorithm and system in an Apple MacBook Pro equipped with Intel Core i7, 4GB DDR3 memory, and an NVidia GeForce GT 330M graphics card with 512MB memory. Interactive visualization performance is achieved. This section describes two experiments to demonstrate the effectiveness and advantages of the seam carving algorithm. A case study is provided to show the usefulness of the system based on the semantic-preserving word clouds for comparing and exploring text documents.

7.1. Experiments

We carried out the first experiment to demonstrate the advantages of our tailored seam carving algorithm over the original seam carving algorithm. In both algorithms, we prevented the seams from passing through the words to ensure word integrity and readability.

We collected *CG&A* abstracts during 2000 to 2001 and created a preliminary sparse layout. Less important keywords were filtered out and only 200 words were retained for simplicity in this experiment. After that, we obtained a very sparse word cloud which was used as the input to both algorithms. We discretized the input sparse word cloud into a 1000×1000 image to run the conventional seam carving technique. Figure 4(a) shows a word cloud packed by a conventional seam carving technique, and Figure 4(b) presents a word cloud generated by our new method. The two figures show that both word clouds are compact and the semantic relations are preserved. By comparing Figure 4(a) and (b), we can see that it is hard to tell the difference. We recorded the time needed for both algorithms in the experiment. The pixel-based seam carving needed 10 seconds to generate the result, while our method needed only 0.5 seconds. This experiment shows that while creating almost identical results, our method is much faster than the pixel-based seam carving technique.

The second experiment was conducted to compare our new method with our previous force-directed algorithm [CWL*10]. We collected 13,828 news articles related to AIG from Jan. 14, 2008 to Apr. 5, 2009. With the data, we created a preliminary word cloud layout (see Figure 5(a)) by multidimensional scaling. We then ran the force-directed algorithm and our seam-carving algorithm on the data, and generated two sequences of word clouds. Figure 5(b) and (c) shows two results for two successive months generated by the force-directed algorithm. Although they contain almost the same keywords (with different sizes) and have the same preliminary layout (i.e., the same semantic structure), the word clouds are quite different. For example, the word groups inside the closed red curves in Figure 5(b) to (c) are quite different (the shapes of the closed curves are rather different). This means that the force-directed algorithm is not linear (or, in other words, not stable) to the input. A slight change to the input often results in a very different layout. In contrast, the sequence of the word clouds created by our new method are more consistent. The same word groups in

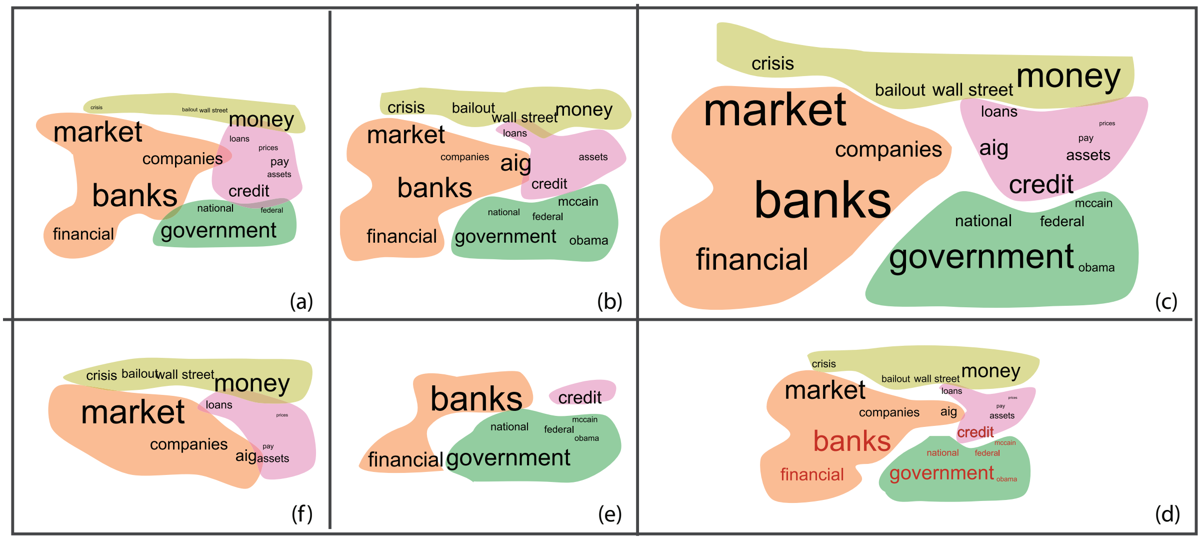
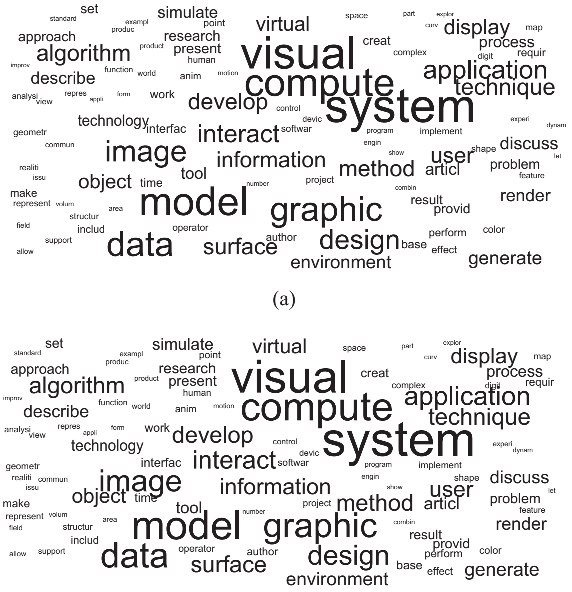


Figure 3: *Illustration for split and merge interactions: (a) and (b) two selected word clouds to be merged; (c) a resulting word cloud created by merging (a) and (b); (d) a word cloud to be split, i.e., a group of keywords in red are selected to be separated from (d); (e) and (f) two resulting word clouds generated by splitting the keywords in (d).*



(a)

(b)

Figure 4: *Comparison of different seam-carving techniques. (a) a word cloud generated by seam-carving by pixels; (b) a word cloud generated by seam-carving by zones. The two results are almost the same, but the zone-based seam-carving runs much faster than the pixel-based technique.*

the yellow bubble set in Figure 5(d) and (e) are similar. This would be easier for users to track and compare the document collection.

From Figure 5 we can also observe that our algorithm works much better than the force-directed algorithm for preserving the semantic relations among words. For instance, the preliminary layout (Figure 5(a)) shows that the word “aig" is to the right of the word “insurance", and “company" is to the left of word the “aig". The two spatial relations are different from what they should be in Figure 5(b) and (c). In contrast, these spatial relations are preserved in Figure 5(d) and (e). Finally, by comparing Figure 5(b)&(c) and (d)&(e), we can find that our new results tend to be more regular. This is more space efficient and especially useful for comparative visualization using the spreadsheet for semantic-preserving word clouds.

7.2. Case Study

This section describes a case study to demonstrate the usefulness of our semantic-preserving word clouds for comparative visualization. We used all the abstract data from IEEE Vis/Infovis and EuroVis from 1999 to 2010 as the input text documents. The goal of this case study was to visually analyze and compare the two conferences, and find out how the conferences evolved over the last decade.

We first extracted all keywords from all the papers, constructed a dissimilarity matrix, and created a preliminary

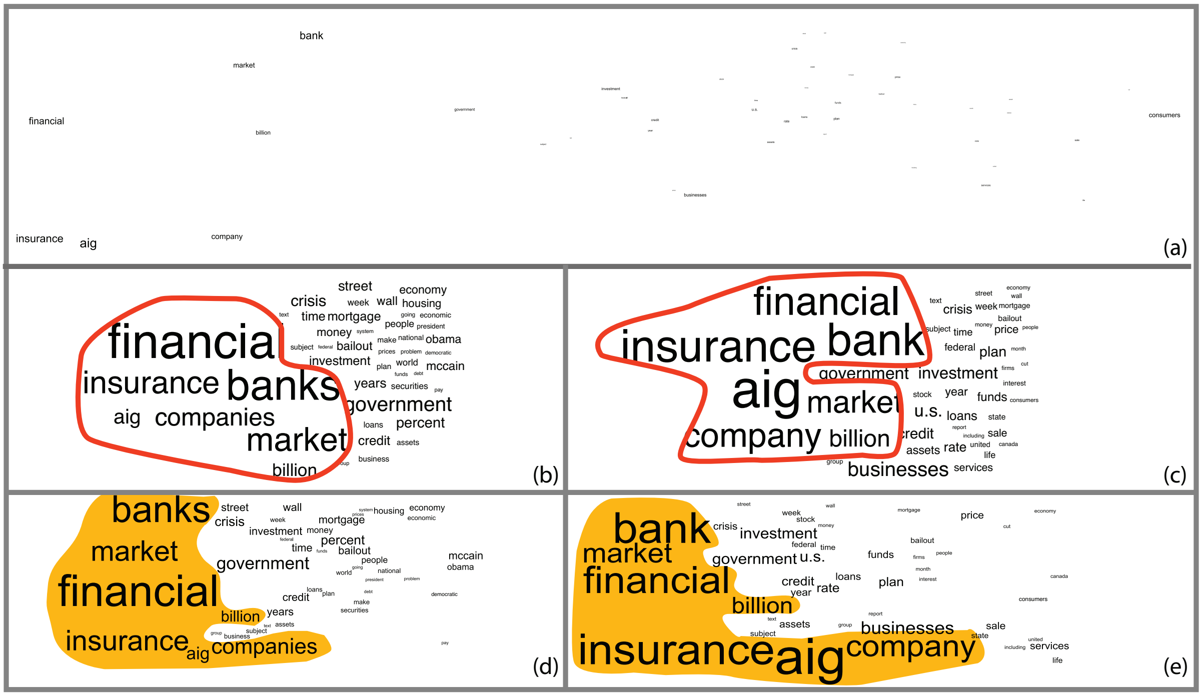


Figure 5: *Comparison of two algorithms. (a) a preliminary sparse word cloud layout for the AIG data; (b)-(c) word clouds created by our previous work [CWL*10]; (d)-(e) word clouds created by our seam carving algorithm.*

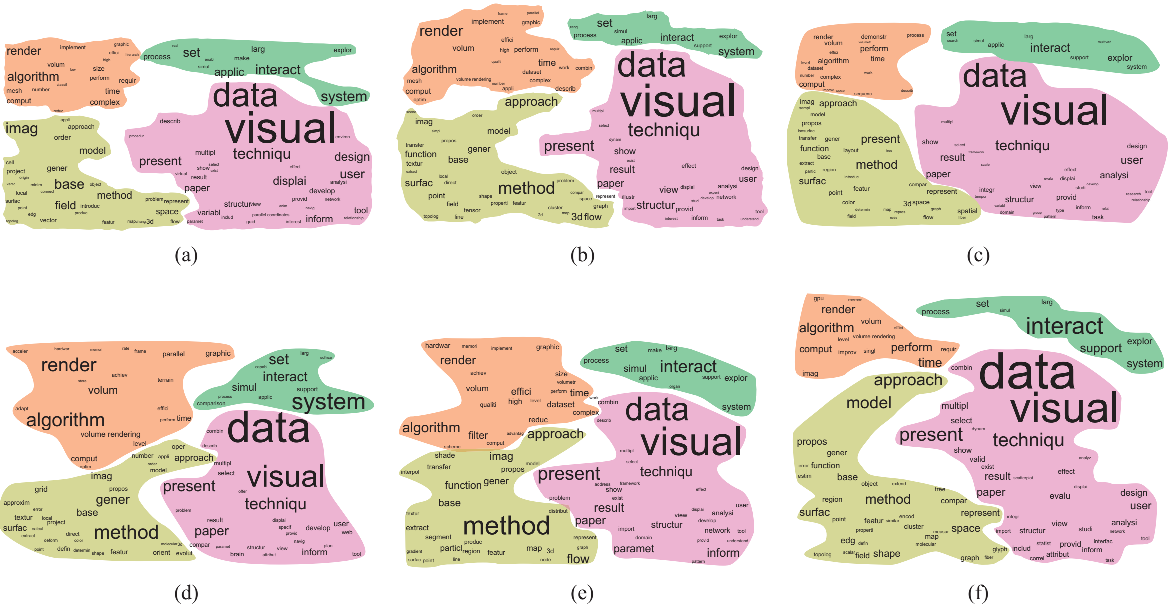


Figure 6: *(a)-(c) semantic-preserving word clouds of the IEEE Vis/InfoVis paper abstracts at 1999, 2005, and 2010; (d)-(f) semantic-preserving word clouds of the EuroVis paper abstracts at 1999, 2005, and 2010.*

word layout L by multidimensional scaling. For every conference i at each year, we removed the irrelevant words that do not appear from L and obtained a sparse layout L_i, and finally generated a semantic-preserving word cloud by removing appropriate seams repeatedly from L_i. Figure 6(a)-(c) shows the word clouds for IEEE Vis/EuroVis at 1999, 2005, and 2010, while Figure 6(d)-(f) shows those for EuroVis at 1999, 2005, and 2010. The bubble sets help users intuitively distinguish groups of keywords. From the created word clouds, we can easily observe the general evolution trend of both conferences by simply looking at how the bubble sets in these word clouds change over time. Generally speaking, the two conferences had very similar groups of keywords (or topics), and have evolved quite similarly in the last decade. “Data" and “Visual" had been always the most important keywords with high occurrence frequency in the paper abstracts in both conferences. The keywords related to rendering performance such as “render", “time", and “computing" became less and less important in both conferences too (i.e., the orange group at the top left regions in the word clouds became smaller and smaller from (a) to (c), and from (d) to (f)). We can also see that “Volume rendering" was a hot keyword from 2000-2005 and became less important after 2005 in both conferences. This case study has demonstrated the usefulness of our techniques for comparing and tracking multiple documents using semantic-preserving word clouds.

7.3. Discussion

The experiments have demonstrated the effectiveness and usefulness of our new algorithm. Nevertheless, our technique may not be useful for all text visualization applications. For example, Wordle would probably be better than our algorithm in applications where aesthetic and compactness aspects are more important. On the other hand, the word clouds created by our method would be more appropriate for analysis and comparison of documents. Our algorithm has some limitations. Our spreadsheet interface is just a proof of concept for demonstrating the feasible use of our semantic-preserving word clouds in comparative visualization. Given a certain screen resolution, the spreadsheet visualization may not scale well with an increasing number of word clouds, as the word readability may become a serious issue when a small word cloud is presented. This issue could be addressed by a multi-resolution technique. We can use bubble sets to encode the quantitative information (the word occurrence) of a group of keywords for providing users with an overview. More details can be shown, when users filter out irrelevant information. Furthermore, interaction techniques such as “search", “split", and “merge" enable users to narrow down their search, allowing for interactive exploration of larger datasets.

Our word clouds usually show only a limited number of keywords. Showing too many words in a word cloud is ineffective because human perceptual capability does not scale well with the increasing amount of information. In our system, users can set the desired number of words and get the resulting word clouds interactively. Finding the upper limit on the number of keywords automatically and designing a perceptually effective word cloud requires further study.

Our work utilizes some keyword extraction techniques such as Porter Stemmer and LexRank widely used in text mining. Nevertheless, these techniques may have some limitations. For example, the stems produced by porter stemmer such as "techniqu" and "provid" do not look nice and might irritate users. We plan to use lemmatization [Lem] to find more compound nouns. We also want to improve the keyword extraction results by Topic Models (such as Latent Dirichlet Allocation (LDA) [BNJ03]) that consider topic information in keyword extraction.

We perform k-means clustering in the 2D preliminary word layouts rather than the high dimensional vector space. Although high-dimensional clustering can give us a hint about how good the preliminary word layouts are, it has proven to be difficult and not intuitive for visualization [KKZ09]. Therefore, we do the clustering in the 2D projection space rather than in the original high dimensional data space. Our technique requires that users specify a “k" value in k-means for clustering an initial word layout. Users can interactively change the “k" value until they obtain their desired results. We plan to use some advanced approaches such as [PM00] to estimate the “k" value automatically in future.

8. Conclusions and Future Work

This paper presents a new algorithm for creating semantic-preserving and compact word clouds using an adapted seam carving technique. With the created semantic-preserving word clouds, we present a spreadsheet visualization interface which allows users to visually compare and explore a document collection interactively and efficiently.

In the future, we plan to conduct a formal user study to verify the intuitiveness of our semantic-preserving word clouds for comparing documents and tracking content evolution. We also want to improve our work by providing a visual indication of similarities and differences, thus allowing for semi-automatic comparison of word clouds. This paper shows one simple application of our technique to compare paper abstracts in two different conferences. Applying our technique to other text analysis and comparison applications such as business analysis and customer opinion analysis is another future direction.

Acknowledgments

This research was supported in part by the HP Labs, AT&T Labs Research, and U.S. National Science Foundation through grants CCF-0808896, CNS-0716691, CCF 0811422, CCF 0938114, and CCF-1025269.

References

[AS07] Avidan S., Shamir A.: Seam carving for content-aware image resizing. *ACM Transactions on Graphics 26*, 3 (2007), Article No. 10. 2, 4

[BNJ03] Blei D. M., Ng A. Y., Jordan M. I.: Latent dirichlet allocation. *The Journal of Machine Learning Research 3*, 1 (2003), 993Ð–1022. 9

[CPC09] Collins C., Penn G., Carpendale S.: Bubble sets: revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1009–1016. 2, 6

[CSL*10] Cao N., Sun J., Lin Y.-R., Gotz D., Liu S., Qu H.: Facetatlas: Multifaceted visualization for rich text corpora. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1172–1181. 2

[CVW09] Collins C., Viégas F. B., Wattenberg M.: Parallel tag clouds to explore and analyze faceted text corpora. In *IEEE Visual Analytics Science and Technology* (2009), pp. 91–98. 2

[CWL*10] Cui W., Wu Y., Liu S., Wei F., Zhou M. X., Qu H.: Context preserving dynamic word cloud visualization. In *IEEE Pacific Visualization Symposium* (2010). 1, 2, 3, 6, 8

[DGWC10] Dörk M., Gruen D., Williamson C., Carpendale S.: A visual backchannel for large-scale events. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1129–1138. 2

[ER04] Erkan G., Radev D. R.: Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res. 22*, 1 (2004), 457–479. 2, 3

[FHRH08] Fisher D., Hoff A., Robertson G., Hurst M.: Narratives: A visualization to track narrative events as they develop. In *IEEE Visual Analytics Science and Technology* (2008), pp. 115–122. 2

[HHWN02] Havre S., Hetzler E., Whitney P., Nowell L.: Themeriver: visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics 8*, 1 (2002), 9–20. 2

[KKZ09] Kriegel H.-P., Kröger P., Zimek A.: Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data 3*, 1 (2009), 1Ð–58. 9

[KLKS10] Koh K., Lee B., Kim B., Seo J.: Maniwordle: Providing flexible control over wordle. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1190–1197. 2

[Lem] Lemmatizer.org Ñ an open-source lemmatizer of english and russian languages. http://lemmatizer.org/. 9

[LRKC10] Lee B., Riche N. H., Karlson A. K., Carpendale S.: Sparkclouds: Visualizing trends in tag clouds. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1182–1189. 2

[LZP*09] Liu S., Zhou M. X., Pan S., Qian W., Cai W., Lian X.: Interactive, topic-based visual text summarization and analysis. In *ACM conference on Information and knowledge management* (2009), pp. 543–552. 2

[Ope] OpenNLP: an organizational center for open source projects related to natural language processing. http://incubator.apache.org/opennlp/. 3

[OST*10] Oesterling P., Scheuermann G., Teresniak S., Heyer G., Koch S., Ertl T., Weber G. H.: Two-stage framework for a topology-based projection and visualization of classified document collections. In *IEEE Visual Analytics Science and Technology* (2010), pp. 91–98. 2

[PG10] Paul M., Girju R.: Comparative scientific research analysis with a language-independent cross-collection model. In *El XXV Congresso de la Sociedad Española para el Procesamiento del Lenguaje Natural* (2010). 2

[PM00] Pelleg D., Moore A.: X-means: extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning* (2000), pp. 727–734. 9

[Por80] Porter M.: An algorithm for suffix stripping. *Program 14*, 3 (1980), 130–137. 3

[RBC*09] Rose S., Butner S., Cowley W., Gregory M., Walker J.: Describing story evolution from dynamic information streams. In *IEEE Visual Analytics Science and Technology* (2009), pp. 99–106. 2

[Ren94] Rennison E.: Galaxy of news: An approach to visualizing and understanding expansive news landscapees. In *UIST' 94* (1994), pp. 3–12. 2

[Sch98] Schütze H.: Automatic word sense discrimination. *Computational Linguistics 24*, 1 (1998), 97Ð–123. 3

[SGL08] Stasko J., Gorg C., Liu Z.: Jigsaw: Supporting investigative analysis through interactive visualization. *Information Visualization 7*, 2 (Summer 2008), 118–132. 2

[SOR*09] Strobelt H., Oelke D., Rohrdantz C., Stoffel A., Keim D. A., Deussen O.: Document cards: A top trumps visualization for documents. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1145–1152. 2

[Sta] The stanford parser: A statistical parser. http://nlp.stanford.edu/software/lex-parser.shtml. 3

[SWL*10] Shi L., Wei F., Liu S., Tan L., Lian X., Zhou M.: Understanding text corpora with multiple facets. In *IEEE Visual Analytics Science and Technology* (2010), pp. 99–106. 2

[VWF09] Viegas F. B., Wattenberg M., Feinberg J.: Participatory visualization with Wordle. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1137–1144. 2

[ZG10] Zesch T., Gurevych I.: Wisdom of crowds versus wisdom of linguists - measuring the semantic relatedness of words. *Natural Language Engineering (2010), 16: 25-59 16*, 1 (2010), 25–59.