Context-Preserving, Dynamic Word Cloud Visualization

Weiwei Cui and Yingcai Wu ▪ *Hong Kong University of Science and Technology*

Shixia Liu, Furu Wei, and Michelle X. Zhou ▪ *IBM China Research Lab*

Huamin Qu ▪ *Hong Kong University of Science and Technology*

The proposed method uses context-preserving, dynamic word clouds to illustrate content evolution. It generates a sequence of word clouds in which related words are grouped together. This sequence is then coupled with a trend chart that summarizes content changes so that users can better explore large collections of documents.

Word clouds provide a concise yet fun way to summarize the content of websites or text documents. So, they're popular in both websites and text analysis systems. In a typical word cloud, tags from a website (or words from a document) are packed into a rectangular region in which font size indicates tag popularity (or word frequency) and font color indicates other useful information (for example, see Flickr www.flickr.com/photos/tags). Recently, researchers have explored alternative layout methods. Some methods cluster words on the basis of their semantic relations (for example, co-occurrence relations),[1] whereas others address aesthetic issues such as eliminating large white space[2] or adhering to specific boundaries.[3]

Although existing layout techniques have achieved some success, they're inadequate at balancing semantically meaningful clusters with visually appealing layouts. For example, one popular layout technique leverages dimensionality reduction methods, such as multidimensional scaling, to project words onto 2D space.[4] However, this technique usually results in wasted empty space; as a result, zooming out to see the overview of a cloud renders words illegible. It would be better to utilize the empty space, making the text bigger and easier to read. Yet arranging clusters in lines[1] to more efficiently use screen space sacrifices relationships between words, and aesthetically appealing word clouds that pack tags more tightly lose meaningful word positions. Furthermore, few methods help users easily track the content changes embedded in dynamic word clouds.

To tackle these challenges, we propose a layout method that combines the merits of existing approaches and overcomes their shortcomings by creating dynamic word clouds. Specifically, we designed an algorithm based on geometry meshes and an adaptive force-directed model that generates word clouds. Our method ensures semantic coherence and spatial stability, which in turn preserves semantic context and makes tracking content changes in word clouds easier. So, our solution is especially suitable for visualizing dynamic contents.

System Overview

Suppose we want word clouds to demonstrate how news topics regarding a specific company (for instance, Apple) varied over the past decade. We could extract keywords from news articles for each month, apply our layout method on the extracted words month by month, and explore the resulting word cloud sequence. (Figures 1a to 1e show five such word clouds for Apple.) We first examine the whole sequence to get the big picture and then identify particularly interesting words and track how those words change in size (in other words,

 Published by the IEEE Computer Society

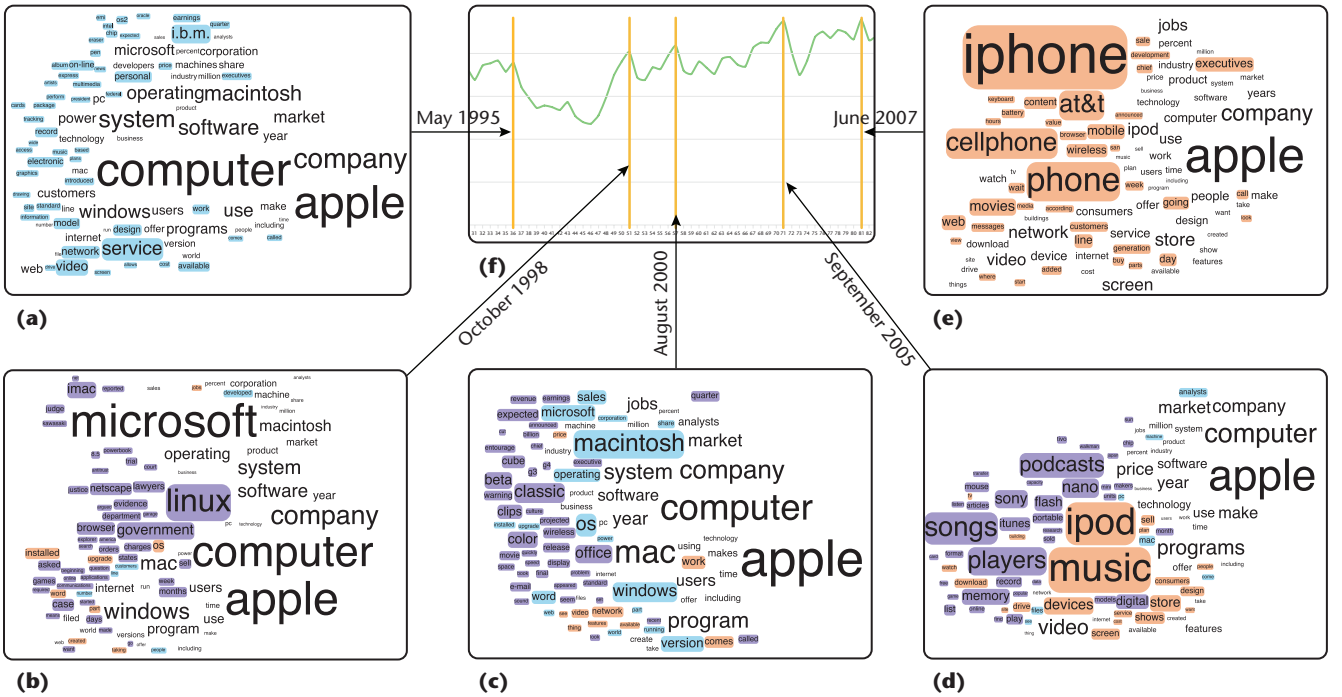

Figure 1. An overview of our system for generating dynamic word clouds. (a–e) Our method creates five word clouds for five selected time points with high significance values. (f) The top center box presents a significance trend chart whose significance curve is extracted from a collection of documents with different time stamps.

popularity) throughout the sequence. For example, in the word cloud for October 1998 (see Figure 1b), the word "microsoft" is interesting; we might want to know how its size changes throughout the sequence.

Our method works well for this task: it packs words tightly, creating a space-efficient overview, and can stabilize word placement across different word clouds (for example, "microsoft" is in the same position in Figures 1a to 1c). In contrast, current layout algorithms generate each word cloud independently, which means positions of "microsoft" in different word clouds might differ dramatically, making it difficult to track.

In some scenarios, tracking individual words might not be adequate for users to correctly (or fully) comprehend what the sequence reveals. Because single words are often ambiguous, we generally need groups of correlated words to derive concrete meaning. Because our solution places words according to their semantic relationships, users can better derive that meaning. For example, we can easily tell that "memory" in Figure 1d refers to "iPod storage" rather than "computer memory."

Going through an entire sequence to find interesting word clouds or words can make exploration tedious. So, we couple a trend chart (see Figure 1f) with the word cloud sequence to illustrate content evolution. The chart lets us see the document content's evolution, which is modeled by various significance values we afford to individual word clouds and is represented by a set of word clouds. Intuitively, significance values describe how documents differ from their neighbors in the temporal domain. To our knowledge, this is the first method that creates context-preserving word cloud visualizations to depict evolving text content.

Our method offers two unique benefits. First, its dual-level visualization illustrates temporal content evolution at different levels and in detail. Second, its time-varying word cloud layout manages both space efficiency and word position stability to help users perceive content changes across clouds.

Significance Analysis

The more information a word cloud conveys and the less information it shares with others, the more significance we afford it—similarly to Chaoli Wang and his colleagues.[5] (For more on this and other research related to word clouds, see the sidebar.) To make our significance analyses, we use entropy to estimate the inherent information in a word cloud, and we use mutual entropy to estimate information shared between two clouds. From these two, we adopt conditional entropy, which quantitatively estimates a word cloud's significance value. For background on information theory, see Wang and his colleagues' paper.[5]

Entropy Estimation

To quantify a word cloud's information entropy $H(X)$, we need a feature vector to represent each

Related Work on Word Clouds

Here we look at three areas of previous research: static word cloud visualization, dynamic word cloud visualization, and entropy in visualization.

Static Word Cloud Visualization

Most static word cloud visualization addresses common aesthetic issues (for example, reducing empty space between words and avoiding overlap) or conveys semantic relations (for example, co-occurrence relations). Christin Seifert and her colleagues developed a family of algorithms that inscribe words into arbitrary convex polygons with little white space.[1] Emden Gansner and Yifan Hu proposed a grid-based algorithm that removes node overlap and preserves proximity relations between nodes.[2] Fernanda Viégas and her colleagues invented Wordle, which uses a greedy placement method to avoid overlap and efficiently use the typographical space.[3] Yusef Hassan-Montero and Víctor Herrero-Solana developed a clustering algorithm based on word co-occurrence information that visually conveys the strength between tags in a word cloud.[4] Unlike these methods, ours aims to analyze content evolution in a stream of text documents using dynamic word clouds.

Dynamic Word Cloud Visualization

Unlike static word cloud visualization, dynamic word cloud visualization is relatively new and addresses a different problem—illustrating content evolution in a stream of documents. Micah Dubinko and his colleagues proposed a tool to visualize the evolution of photo tags on Flickr, in which users observe and interact with words as they evolve over time.[5] Compared with this method, which generates an animation of word evolution, ours provides a significance trend chart that depicts the variation of word clouds over time. So, users get a visual overview of the varying trends in the clouds. Also, our method employs a geometry-based method that generates word cloud layouts so as to ensure their semantic coherence and spatial stability over time.

Christopher Collins and his colleagues introduced parallel tag clouds (PTCs), which exploit both parallel coordinates and traditional tag clouds.[6] PTCs visualize the content evolution of a stream of documents; each PTC column shows the important words from document collection at a certain time point. However, our method is more intuitive in providing such an overview and doesn't require parallel-coordinate expertise.

Entropy in Visualization

Entropy was first introduced in thermodynamics to measure the disorder in a system. Since then, it's been extended to information theory to describe random processes' long-term behavior. Recently, various visualization systems have adopted entropy to deal with large or time-varying data. For example, Chaoli Wang and his colleagues used entropy to visualize large, time-varying volumetric data.[7] In their method, the more exclusive information that volume data has at a time point (versus information shared with others), the more significant that data is. This has inspired our entropy approach, which also considers word clouds more significant when they contain more exclusive information.

Kamel Aouiche and his colleagues also adopted entropy to measure the disparity of font sizes between words in a word cloud.[8] Their research focuses on a single word cloud and quantifies how interesting (or irregular) a word frequency pattern is in the cloud. In contrast, our method measures how significant (or different) a word cloud is in a whole word cloud sequence.

References

1. C. Seifert et al., "On the Beauty and Usability of Tag Clouds," *Proc. Int'l Conf. Information Visualization*, IEEE Press, 2008, pp. 17–25.
2. E. Gansner and Y. Hu, "Efficient, Proximity-Preserving Node Overlap Removal," *J. Graph Algorithms and Applications*, vol. 14, no. 1, 2010, pp. 53–74.
3. F.B. Viégas, M. Wattenberg, and J. Feinberg, "Participatory Visualization with Wordle," *IEEE Trans. Visualization and Computer Graphics*, vol. 15, no. 6, 2009, pp. 1137–1144.
4. Y. Hassan-Montero and V. Herrero-Solana, "Improving Tag-Clouds as Visual Information Retrieval Interfaces," *Proc. Int'l Conf. Multidisciplinary Information Sciences and Technologies*, ACM Press, 2006, pp. 25–28.
5. M. Dubinko et al., "Visualizing Tags over Time," *ACM Trans. Web*, vol. 1, no. 2, 2007, pp. 1–22.
6. C. Collins, F.B. Viégas, and M. Wattenberg, "Parallel Tag Clouds to Explore and Analyze Faceted Text Corpora," *Proc. IEEE Symp. Visual Analytics Science and Technology* (VAST 09), IEEE Press, 2009, pp. 91–98.
7. C. Wang, H. Yu, and K.-L. Ma, "Importance-Driven Time-Varying Data Visualization," *IEEE Trans. Visualization and Computer Graphics*, vol. 14, no. 6, 2008, pp. 1547–1554.
8. K. Aouiche, D. Lemire, and R. Godin, "Web 2.0 OLAP: From Data Cubes to Tag Clouds," *Web Information Systems and Technologies*, Springer, 2009, pp. 51–64.

word in the cloud from multiple perspectives. In our system, the feature vector consists of three elements (f, x, y) that correspond to the word frequency f in the text and the word positions (x, y) in the cloud. How we choose the feature vector is important because we use it to represent information in a word cloud as well as changes between word clouds. No doubt word frequency is the most

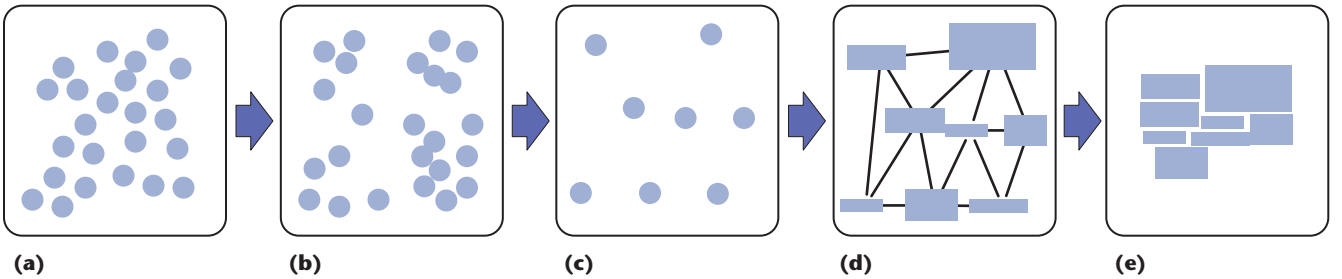

Figure 2. The pipeline for creating a semantic and stable word cloud layout. (a) Extracting an initial set of words from documents with different time stamps. (b) Placing extracted words on a 2D plane using multidimensional scaling. (c) Filtering out unrelated words for a specified time point. (d) Triangulating the remaining words. (e) Optimizing the layout by a force-directed algorithm.

important piece of information to include; however, knowledge about word position is also necessary when describing word clouds. So, combining the two pieces of information is a comprehensive, reasonable solution.

We can build a multidimensional histogram on the feature vectors. To strike a balance between performance and storage demand, we set the number of intervals for each dimension in our histograms to 32. Each bin in the histogram counts the number of words that fall into certain disjoint feature value intervals. We compute $H(X)$ by the normalized count of values in each bin. Mathematically, the information entropy of word cloud X is

$$H(X) = -\sum_{f \in F} \sum_{x \in X} \sum_{y \in Y} p(f,x,y) \log p(f,x,y).$$

F, X, and Y are bin sets in those three dimensions; $p(f, x, y)$ represents the normalized count of values falling in (f, x, y).

Mutual-Entropy Estimation

Given word clouds X and Y, the information shared between them are the words common to both clouds. We consider the words in one cloud but not the other to be independent. So, we construct the joint histogram by counting the number of words that fall into a particular interval of a combination of the feature values, including the word frequency, position, and color values of one word cloud. We then compute the mutual information $H(X; Y)$ by the normalized count of every bin of the joint histogram. In other words, we compute $H(X; Y)$ by $p(x, y)$ as well as $p(x)$ and $p(y)$ as measured on the basis of the multidimensional histogram in the previous section.

Conditional-Entropy Estimation

To estimate the conditional entropy of X at time t, we consider several preceding or succeeding word clouds. This means we choose word clouds of time points in a given window centered at t. Supposing t_i is the weight of word cloud Y_i in the window and *Size* is the window size, we define the significance of X as

$$S(X) = \sum_{i=1}^{Size} t_i \cdot H(X|Y_i) = \sum_{i=1}^{Size} t_i \cdot \left(H(X) - H(X;Y_i)\right).$$

Here, the summarization of t_i is one: $\sum_i^S t_i = 1$.

Word Cloud Layout

Although word clouds are more flexible than standard time series views in showing complex word relationships, they're not typically designed for side-by-side comparison. For instance, a word's size and position usually varies across different word clouds, and words frequently appear or disappear over time. So, with conventional word clouds, exploring temporal patterns of documents with different time stamps is difficult. We introduce a flexible method to create word cloud layouts specifically for such documents. Our method can organize layouts according to different semantic-coherence criteria (which we describe in detail later), to meet different user requirements.

Figure 2 shows our pipeline for creating word cloud layouts for documents with different time stamps. The pipeline begins with an initial set of words extracted from a collection of documents (see Figure 2a). Then we place all extracted words on a 2D plane on the basis of their attributes (see Figure 2b). For a chosen time point, our system filters out all unimportant or unrelated words from the 2D plane (see Figure 2c) and performs Delaunay triangulation[6] on the remaining points, each of which is at the center of a word, to generate a triangular mesh. We determine each word's font size by the corresponding word frequency at that time point (see Figure 2d). Finally, we apply an adapted force-directed algorithm to adjust point positions and obtain an appropriate layout (see Figure 2e).

Word Extraction

Consider n documents $T = \{T_1, T_2, ..., T_n\}$ with different time stamps. For document T_i, we first

remove the most common unimportant words, such as "a," "the," "that," and "thus." Our system then builds a histogram $Hist_i$ to indicate the frequency of all unique words in T_i. Meanwhile, we employ the Porter stemming algorithm to combine words with the same root—for instance, we might group "fishing," "fish," "fisher," and "fished." Because presenting all words to users is usually unnecessary, we select words remaining in $Hist_i$ with frequencies higher than a user-specified threshold as candidate word set W_i, which creates the word cloud for T_i. Finally, we obtain a set of extracted words $W = \{W_1, W_2, ..., W_n\}$ for T.

Semantic clustering lets users understand and track major content efficiently; rather than examining all words one by one, users can quickly look at word clusters.

Initial Word Placement

With the extracted word set W, we place all important words (in other words, $\cup W$) on a 2D plane to create an initial word layout that semantically groups words. This improves word clouds' readability because we organize them into semantically coherent clusters rather than in alphabetical order. Semantic clustering lets users understand and track major content efficiently; rather than examining all words one by one, users can quickly look at word clusters.

To meet different user requirements, our system follows the following three semantic-coherence criteria to generate different layout styles. In conjunction with these criteria, we've established three types of feature vectors to aid in clustering words.

The *importance criterion* creates layouts that cluster words on the basis of importance values at different time points. In other words, because font sizes represent importance values, this criterion groups words with similar variations in font sizes over time. So, important words will appear together. The corresponding feature vector is $V_i = \{v_1, v_2, ..., v_n\}$, where n is the number of time points in the documents and v_j is the importance value (the font size) of word wd_p at time point j.

The *co-occurrence criterion* ensures that words with similar appearances or disappearances over time are clustered together. That is, words appearing or disappearing simultaneously will likely be grouped together. So, co-occurring words will be updated simultaneously. The corresponding feature vector is $V_a = \{v_1, v_2, ..., v_n\}$, with $v_j = 1$ if wd_p is visible at j, and $v_j = 0$ otherwise.

The *similarity criterion* creates layouts in which semantically similar words are clustered. To define semantic similarity between words, we employ Hinrich Schütze's well-established method,[7] which suggests that semantically similar words share similar neighboring words. The feature vector is $V_s = \{v_1, v_2, ..., v_m\}$, where m is the number of words in $\cup W$. The element v_q represents the number of times $wd_q \in \cup W$ occurs close to wd_p (in a sentence or larger context) in the documents.

We can evaluate the similarity between vectors V_p and V_q by the cosine measure:

$$\cos(\theta) = \frac{V_i \cdot V_j}{\|V_i\| \cdot \|V_j\|} .$$

The higher the cosine's value, the more similar the two corresponding words. For example, the cosine's value is 1.0 if and only if two words share exactly the same characteristics (in other words, the two words are a perfect match).

With the vector representations and the similarity measurement, we create a dissimilarity matrix Δ, where element $\delta_{p,q}$ represents the similarity ($\cos(\theta)$ in the previous equation) between words p and q. With Δ, we then employ multidimensional scaling (MDS)[8] to reduce each high-dimensional vector to a 2D point. This lets us obtain an initial layout that semantically clusters correlated words on the 2D plane.

Delaunay Triangulation

The initial word layout contains all important words ($\cup W$) of the whole document collection. Nevertheless, users might only want to visualize some documents at a specific time point. In this case, our system filters out the unimportant or unrelated words in the initial layout, as we mentioned before. This often creates a sparse layout (see Figure 2c) that wastes a significant amount of space. To reduce the space between the remaining words, we need to pack them in the layout. On the other hand, the semantic relations between the words are represented implicitly by the relative positions between the words. This information is critically important for the analysis of the documents. So, the packed layout should preserve the relative positions.

We achieve this by using a triangle mesh as the control skeleton to maintain the original relative positions. We perform Delaunay triangulation[6] on the word positions to obtain the mesh, which

we denote as an initial graph $G = (V, E)$. With the graph, we can rearrange the word positions on the 2D plane flexibly to reduce empty space while keeping the semantic relations between the words.

Force-Directed Model

With G, we build a compact word cloud layout. We propose an adapted force-directed algorithm to reposition the vertexes V in G and to remove empty space. Although this process loses information regarding the distance between words, it largely preserves semantic relationships because the topology of G (which encodes underlying semantic word relationships) remains unchanged. To ensure an appropriate force-directed algorithm that maintains graph topology while removing most empty space between words, we follow three design principles.

The *overlapping principle* requires that words don't overlap. It has top priority over the other principles and guarantees each word's readability in the resulting layout.

The *planar principle* ensures that the controlling mesh (the initial graph G) stays as planar as possible. It helps word clouds maintain semantic relationships between words. This principle has lower priority than the overlapping principle. It doesn't need to be strictly followed because keeping semantic relationships doesn't necessitate a strictly planar mesh and because keeping the mesh strictly planar might lead to an unnecessary waste of space.

The *compact principle* removes empty space between words as much as possible so that the created layout is compact. This principle has the lowest priority.

Following these principles ensures the word cloud is easy to read, stable, semantically meaningful, and compact. The model has three basic forces corresponding to the three principles. The *repulsive force* corresponds to the overlapping principle and prevents a word from being occluded by other words. The force takes effect between two words if and only if they overlap each other. The repulsive force f_r is

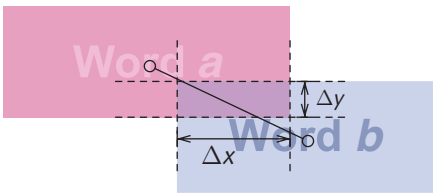

Figure 3. Two overlapped words exert a repulsive force on the connected edge. As long as these two words are overlapping, the repulsive force will always exist and finally make sure they are separated.

$$f_r(a,b) = \begin{cases} k_r \min(\Delta x, \Delta y) & \text{if word } a \text{ overlaps word } b \\ 0 & \text{otherwise,} \end{cases}$$

where k_r is a given weight and Δx and Δy are the overlapping region's width and height (see Figure 3).

The *attractive force* ensures the layouts are stable and semantically meaningful; it corresponds with the planar principle. During layout adjustment, if a mesh triangle is flipped (in other words, if one vertex in the triangle goes to the other side of its subtense), the mesh will become nonplanar. In this case, the attractive force between the subtense and the vertex takes effect and flips the triangle back (see Figure 4). The force f_a is

$$f_a(a,l) = \begin{cases} k_a \Delta d & \text{if word } a \text{ is flipped} \\ 0 & \text{otherwise,} \end{cases}$$

where k_a is a given weight and Δd is the distance between word a and its subtense e.

The *spring force* removes empty space and packs

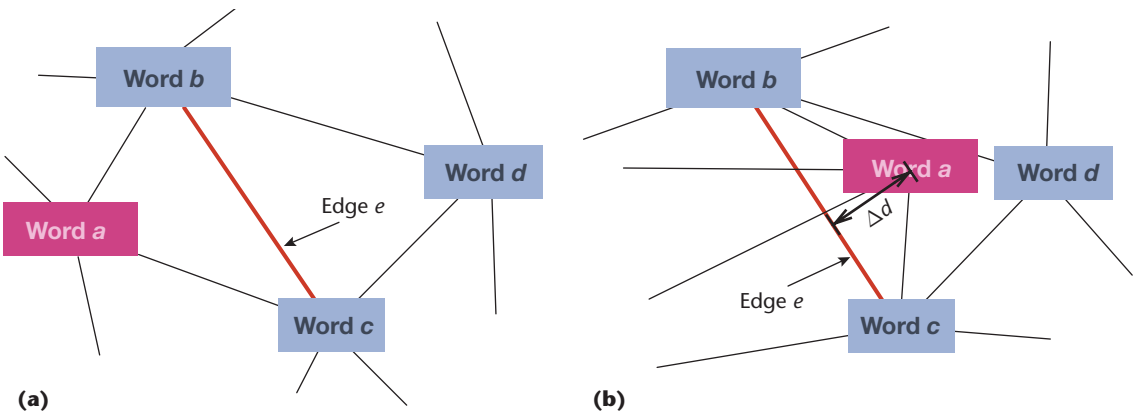

Figure 4. The attractive force ensures layouts are stable and semantically meaningful. (a) The force between edge *e* and word *a* is zero if the mesh is planar. (b) The force takes effect if *a* is flipped to the other side of *e*.

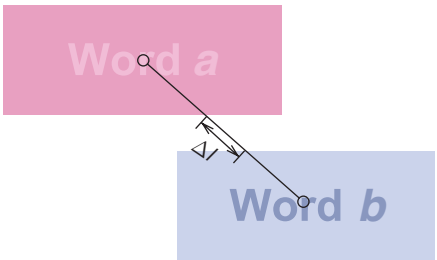

Figure 5. Two separated words exert a spring force on the connected edge. The spring force will try to draw the words to each other, and finally make a compact layout.

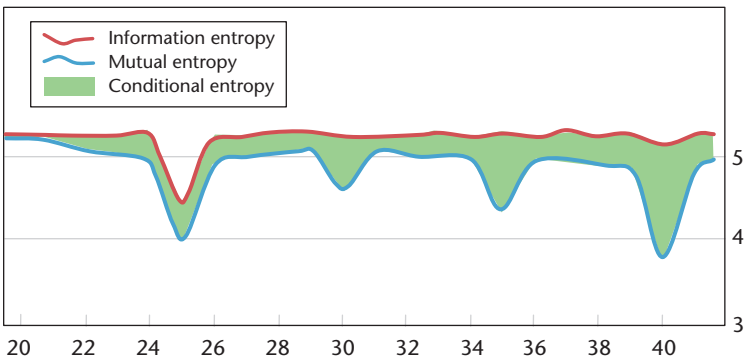

Figure 6. An expanded significance trend chart. With three types of entropies shown at the same time, users may gain better insight and find more interesting moments for further exploration.

words compactly in correspondence with the compact principle. Suppose words a and b are connected in G; the spring force between them is

$$f_s(a,b) = w_a w_b \Delta l, \tag{6}$$

where w_a and w_b are the importance values of words a and b, and Δl represents the length of the connected edge that lies outside both a and b (see Figure 5).

Because the three forces have different priorities, we should choose k_r and k_a according to the design principle priorities. For example, we can set $k_r \ll k_a \ll w_{\max}^2$, where $w_{\max}$ is the words' maximum importance value.

User Interactions

Our system employs several intuitive interaction methods to facilitate exploration of multiple documents.

Expanding the Significance Curve

The trend chart's significance curve expands to provide more details whenever users click on it. As Figure 6 shows, users can use this feature to gain information on the information entropy, mutual entropy, and conditional entropy.

Creating a Storyboard

Users can selectively visualize specific word clouds by sliding a bar at the bottom of the trend chart. When users click on the chart, a separate window pops up, showing the word cloud at that time point. So, users can create a storyboard by selecting and putting together significant word clouds. Either the system can automatically choose word clouds at time points with a significance curve peak, or the user can manually indicate important word clouds on the curve. For example, in Figure 1, a user clicked five time points on the curve, and five windows opened to show corresponding word clouds.

Anchoring Words

Sometimes a user might be particularly interested in certain words and want to track their changes. Although our layout algorithm generally ensures words don't drift far in different word clouds, there's no guarantee that users will easily find them. So, our system can anchor words in the same position across all word clouds. If users are interested in only certain words in the whole sequence, they can click on only those words so that the system automatically aligns those words' positions in all subsequent word clouds. After that, the system still employs the layout algorithm to generate new layouts but won't change the selected words' position. With all the interesting words fixed and aligned, users can easily track word changes across word clouds.

Displaying Frequency Changes

In exploring documents with varying time stamps, tracking certain keywords' frequency change is a common (sometimes important) task. We overlay a line chart on each keyword in the word cloud such that users can easily perceive both the foreground line charts and the background keywords. The system enables three display modes for the line charts: always show them, show them when the mouse hovers on the keyword, or never show them.

Compared with straightforward methods that present this information in separate windows, our method provides an integrated view that avoids frequent context switches. Of course, when words are too small, the overlaid line chart might be difficult to read. However, users generally focus on larger words anyway, and those interested in small words can always zoom in to improve the line charts' readability.

Case Studies

To demonstrate our system's effectiveness and use-

fulness, we applied it on several datasets using an iMac (3.06-GHz Intel Core Duo CPU and 4 Gbytes of RAM). All time-consuming tasks, such as the significance estimation and initial word placement, take place in a couple of minutes during preprocessing. So, all other output—including the significance curve and the word clouds—can be created interactively.

Artificial City Data

To illustrate layout generation, we manually generated a dataset consisting of eight capital city names such that their positions reflected geographical locations. We set font sizes.

Figures 7a and 7b show the initial layout and its mesh generated by Delaunay triangulation. Figures 7c and 7d show a sparse word cloud and the mesh captured at a step during layout adjustment. Figures 7e and 7f are the final word cloud layout and its mesh.

The initial layout has severe clutter. The figures indicate that the repulsive force first dominated word movement and separated overlapped words (see Figure 7c). However, this made a triangle flip in the mesh (the red area in Figure 7d), so the attractive force took effect and flipped it back. Meanwhile, the spring force effectively packed the separated words. The mesh for the resulting word layout is finally planar (see Figure 7f). All words in Figure 7e are kept in their relative positions as those in Figure 7a.

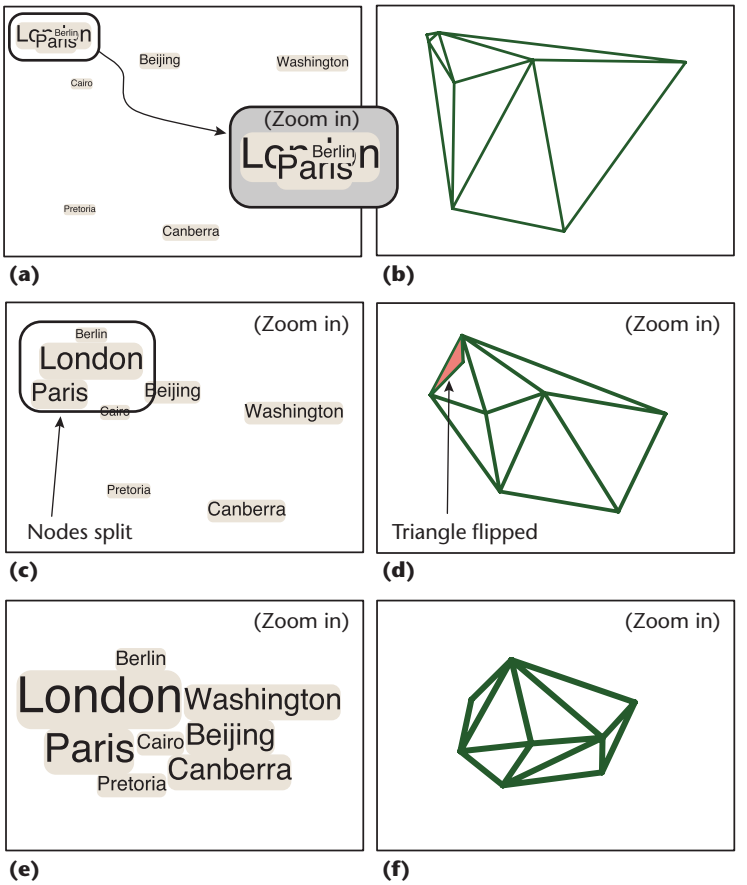

Figure 7. The word cloud layout process: the (a) initial layout and (b) its mesh generated by Delaunay triangulation, a (c) sparse word cloud and (d) its mesh captured at a step during layout adjustment, and the (e) final word cloud layout and (f) its mesh. The figures illustrate key adjustments during the process. We can see that, after the adjustments, city names are packed together while their relative positions are kept.

AIG News Data

We conducted this case study on a real dataset consisting of 13,828 news articles spanning one year (14 Jan. 2008 to 5 Apr. 2009) that were related to American International Group (AIG). Using these articles, we generated a sequence of word clouds. For every cloud, we generated two layouts—one using our method (see Figure 8a) and one using Wordle[2] (see Figure 8b). Regarding compactness, our layouts are comparable to those created by Wordle. However, our layouts have two unique advantages over Wordle. First, our layouts present more semantically meaningful information. For example, looking only at the cluster bordered in blue in Figure 8a, we can easily tell that the underlying documents refer to both the economy and the presidential election. This is because a group of economic words appears together with "Obama" and "McCain." Users might be able to see "Obama" or "McCain" separately (bordered in blue in Figure 8b) in the Wordle layout. However, because the words are far from each other, users might have a difficult time determining the content topic (the US presidential election). So, we can see that information is broken into segments when related words are randomly placed.

Second, because semantically clustered words greatly narrow the visual search space, our method can more efficiently help users compare and track different word clouds over time. For example, users can easily track keyword variations (for example, "economy" and "Obama") between the left and right images in Figure 8a and figure out that "economy" becomes smaller whereas "Obama" disappears. In contrast, users might take longer to do so in Figure 8b because they must search a much larger space. Furthermore, we measured the average offset of all shared words between the two images in Figure 8a and the two images in Figure 8b. The average offset is 122 pixels using our method and 313 pixels using Wordle. These results further indicate that users must search a larger space to track common words when using Wordle.

Additionally, we applied different semantic criteria to the word clouds in Figure 8 to demonstrate

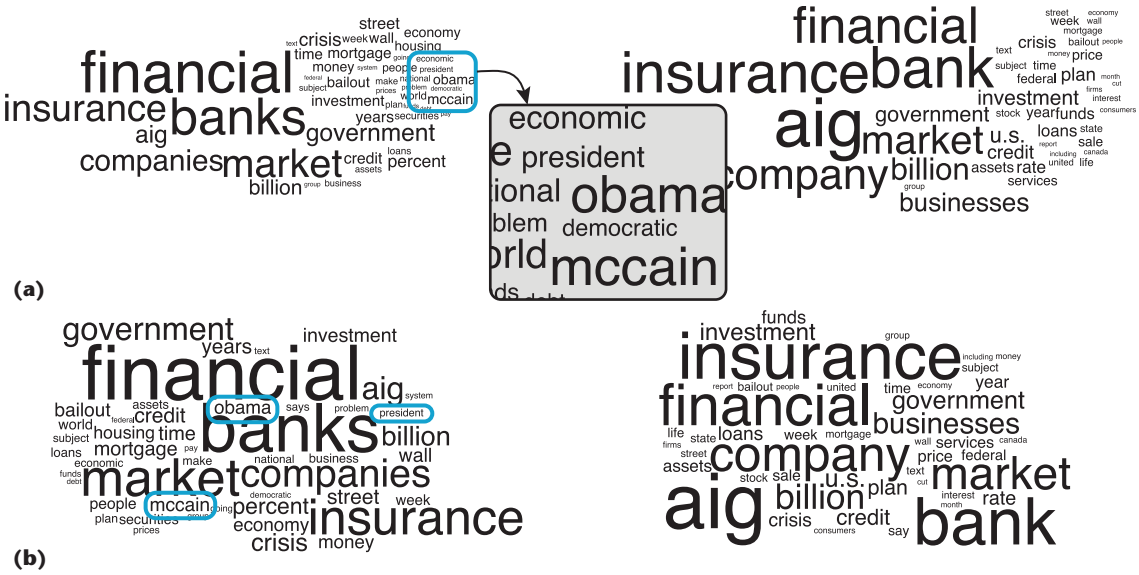

Figure 8. Word cloud layouts created by (a) our method and (b) Wordle. The clouds on the left are generated for documents at the same time point, as are the clouds on the right. Users must search a larger space in the Wordle layouts.

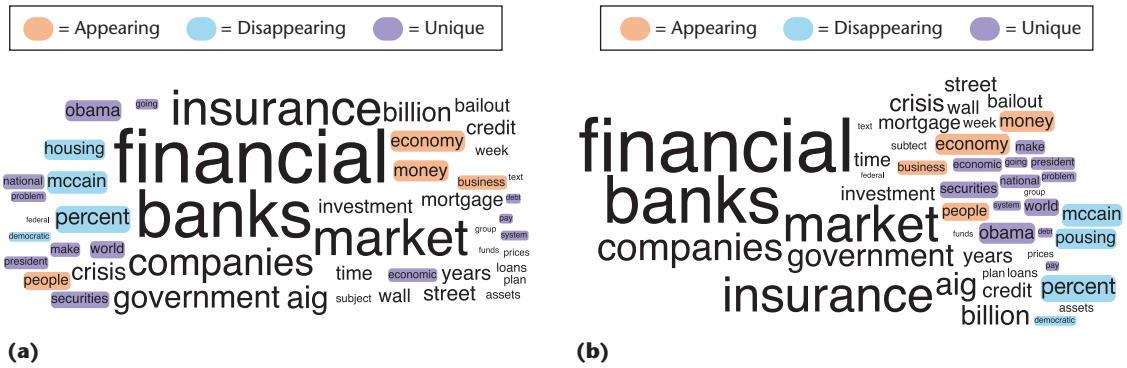

Figure 9. Word cloud layouts generated by the (a) importance criterion and (b) co-occurrence criterion. As we expected, all big words in (a) are placed in the center, while all words with the same background color are generally grouped together in (b).

Figure 10. How positions change as a result of our system's word anchoring. (a) The upper right of Figure 8a. (b) The upper right of Figure 8b. (c) How the upper right of Figure 8b changes when "money" and "funds" are anchored in (a). We can see that the positions of these two words are exactly the same in (a) and (c). Therefore, users should have less trouble tracking them in (c).

different word cloud styles. The word cloud on the left in Figure 8a is generated by the similarity criterion. Figures 9a and 9b present the same data but are generated by the importance criterion and the co-occurrence criterion. The words are highlighted differently according to their appearing behavior. In Figure 9a, words are grouped according to font size; in Figure 9b, words with the same color are roughly clustered together. This experiment shows that our technique success-

fully groups words according to various semantic-coherence criteria.

Furthermore, we used the same dataset to demonstrate word anchoring. Figures 10a and 10b show the upper right of Figures 8a and 8b. The relative positions of "money" and "funds" (bordered in orange) change dramatically between Figures 8a and 8b. However, after we designated these as anchor words for Figure 10a, our system automatically adjusted the layout in Figure 10b to preserve their positions. Figure 10c displays the resulting layout.

CG&A Abstract Data

This case study demonstrates that our system can visually illustrate topic evolution. We first collected 1,984 abstracts from articles published in *IEEE Computer Graphics and Applications (CG&A)* from 1981 to 2009. We then grouped the abstracts by year and generated a word cloud for each year.

Figure 11 shows some results. Our stable layout algorithm lets us easily track and compare sizes of the same keyword between different word clouds. After examining the word clouds side by side, we observed several interesting patterns. Primarily, the word "visualization" continuously grows after it first appears in the 1988 word cloud; the overlaid frequency line chart also reveals this word's increased popularity. Meanwhile, "graphics" decreases continuously in size, indicating that "visualization" has replaced "graphics" as the dominant topic in *CG&A*.

"Surface" also generally decreases in size. However, its size in 2004 is unexpectedly big, given the overall trend (see Figure 11d). After looking into the documents, we found that there were two special issues related to "haptic" that year and that "surface" in this context meant "touch surface." This explains the strange behavior of "surface" as well as the sudden appearance of the huge keyword "haptic" next to "surface" that year. Additionally, some topics emerged only in the last two years, such as "games," "knowledge," and "mobile" (bordered in red in Figure 11e).

Because of the special focus of *CG&A*, we thought it interesting to compare "applications" and "research" (bordered in green in Figure 11). We found that they're both quite stable throughout the years. Although, generally speaking, "applications" is a little bigger than "research," this difference isn't obvious. This suggests that *CG&A* article abstracts don't have preferences between these two words. We also found that "applications," "user," "interactive," and "information" are usually close together (indicating strong correlations) and similar in size. The

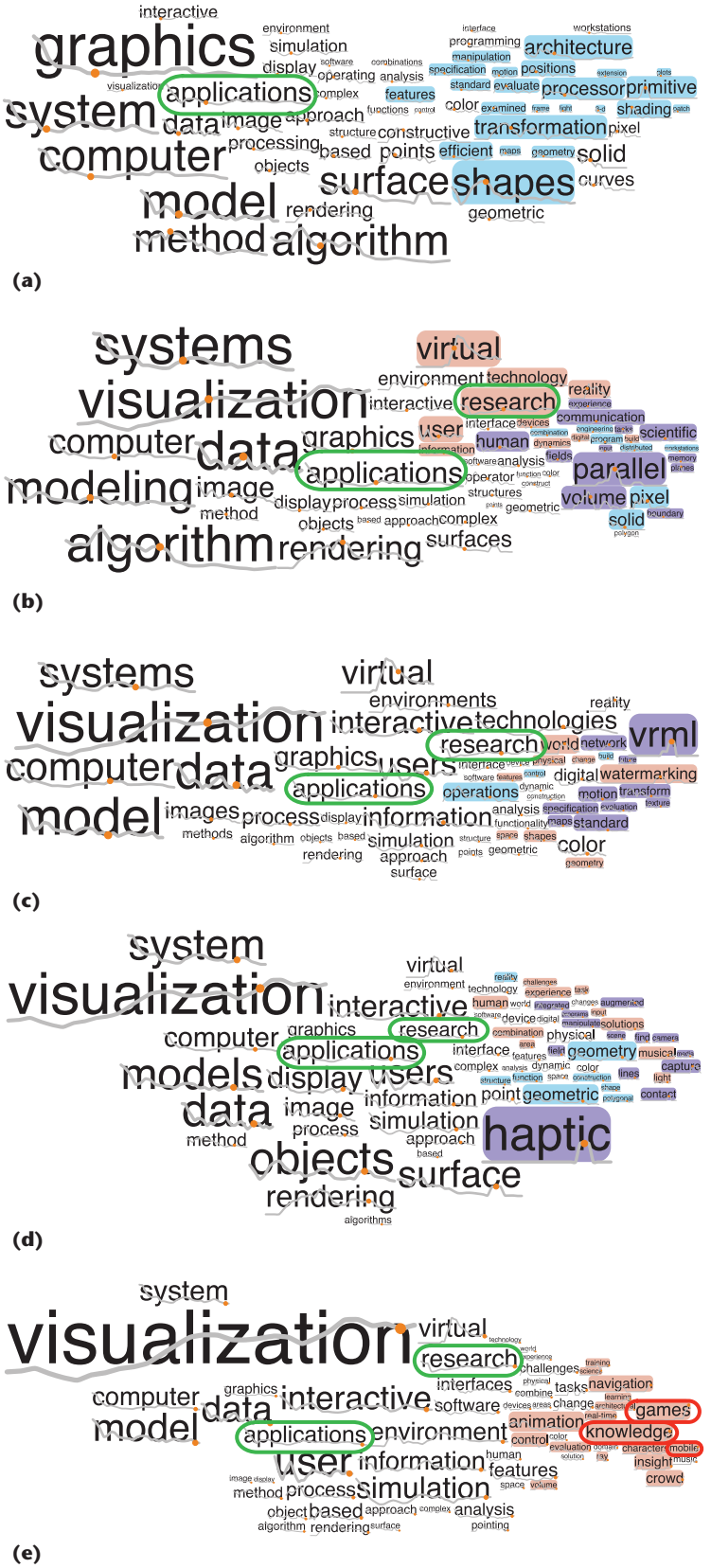

Figure 11. Word clouds summarizing five years of *IEEE Computer Graphics and Applications* article abstracts: (a) 1989, (b) 1994, (c) 1999, (d) 2004, and (e) 2009. In the sequence, some words appear, some words disappear, and some words change in size. By tracking those behaviors, several interesting patterns are observed.

combination of these four words usually indicates an applications topic involving rich user interaction for information exploration.

Apple News Data

This case study was the most comprehensive for demonstrating general usefulness. We carried it out on a collection of 1,933 news articles related to Apple from August 1989 to August 2009. Figure 1f shows the computed significance curve, which presents the news articles' content evolution. Starting in August 1998, the curve is generally higher than it was previously, which coincides with Steve Jobs's return to Apple in late 1996. After Jobs's return, Apple kept itself in the spotlight and maintained high visibility by creating various hot topics. Although the curve is roughly higher on the chart's right (after late 1996), four peaks occur in October 1998, August 2000, September 2005, and June 2007. (These last three peaks roughly match Apple's major product announcements or release dates.)

For comparison, we extracted the four word clouds from these peaks as well as the May 1995 word cloud (which represents the time before Jobs returned to Apple). Figures 1a to 1e clearly summarize Apple's key steps during these years—from computer to iPod to iPhone. Several interesting patterns appear. For instance, thanks to the anchored position of "computer," users can easily see that its size decreases monotonically as Apple's computer gave way to Apple's more popular products, the iPod (released in 2000) and iPhone (released in 2007).

Another interesting pattern is the disappearance of words related to Microsoft from Figures 1d and 1e, which might indicate that Apple broadened its market so successfully the media no longer considered it a Microsoft competitor. In contrast, "microsoft" appears abnormally big in Figure 1b. Upon investigation, we found that an interesting agreement between the two companies took place in October 1998, in which Microsoft invested $150 million in Apple.

Our system also lets us examine semantic relationships between keywords. For example, by comparing the words surrounding "ipod" in Figure 1d and surrounding "iphone" in Figure 1e, we can determine which features people were most interested in regarding the two products ("memory" and "portable" for the iPod, and "keyboard" and "battery" for the iPhone).

This case study's results demonstrate that our method produces effective summaries for large bodies of text.

Because our system groups semantically coherent words to ensure spatial stability over time, our layout works better than others for visualizing documents with time stamps. Our significance estimation of word clouds provides users a visual summary of semantic variations in the document content.

Nevertheless, our method has a few limitations. It usually creates layouts as compact as those generated by existing methods. However, it might not deliver a compact layout when the initial layout is irregular (for example, when most words are placed in a straight line). Setting a higher priority for the spring force and a lower priority for the attractive force alleviates this problem. However, the initial layout depends on semantic information and can't be adjusted manually. We plan to improve this by enabling user interaction and integrating user knowledge into the initial layout generation.

Additionally, although our system lets users create a storyboard from documents, simply selecting word clouds from the significance curve peaks might not tell the whole story. So, we'll also study how to effectively select word clouds for a story presentation.

Acknowledgments

Most of this research occurred while Weiwei Cui and Yingcai Wu were on summer internships at the IBM China Research Laboratory. This research was supported partly by Hong Kong Research Grants Council grant 618706 and General Research Fund grant 619309 as well as an IBM Faculty Award.

References

1. Y. Hassan-Montero and V. Herrero-Solana, "Improving Tag-Clouds as Visual Information Retrieval Interfaces," *Proc. Int'l Conf. Multidisciplinary Information Sciences and Technologies*, ACM Press, 2006, pp. 25–28.
2. F.B. Viégas, M. Wattenberg, and J. Feinberg, "Participatory Visualization with Wordle," *IEEE Trans. Visualization and Computer Graphics*, vol. 15, no. 6, 2009, pp. 1137–1144.
3. C. Seifert et al., "On the Beauty and Usability of Tag Clouds," *Proc. Int'l Conf. Information Visualization*, IEEE Press, 2008, pp. 17–25.
4. K. Fujimura et al., "Topigraphy: Visualization for Large-Scale Tag Clouds," *Proc. 17th Int'l Conf. World Wide Web* (WWW 08), ACM Press, 2008, pp. 1087–1088.
5. C. Wang, H. Yu, and K.-L. Ma, "Importance-Driven Time-Varying Data Visualization," *IEEE Trans. Visuali-*

zation and Computer Graphics, vol. 14, no. 6, 2008, pp. 1547–1554.
6. M. de Berg et al., *Computational Geometry: Algorithms and Applications*, 2nd ed., Springer, 2000.
7. H. Schütze, "Automatic Word Sense Discrimination," *Computational Linguistics*, vol. 24, no. 1, 1998, pp. 97–123.
8. I. Borg and P.J.F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, 2nd ed., Springer, 2005.

Weiwei Cui *is a PhD candidate in computer science at the Hong Kong University of Science and Technology. His research focuses on information visualization, especially graph visualization and text visualization. Cui has a BSc in computer science from Tsinghua University. Contact him at weiwei@cse.ust.hk.*

Yingcai Wu *is a postdoctoral researcher in the Visualization and Interface Design Innovation group at the University of California, Davis. His research interests include visual analytics, information visualization, and medical visualization. Wu has a PhD in computer science from the Hong Kong University of Science and Technology. Contact him at wuyc@cse.ust.hk.*

Shixia Liu *is a lead researcher in Microsoft's Internet Graphics group. Her research interests include information visualization and visual analytics. Liu has a PhD in computer-aided design and computer graphics from Tsinghua University. Contact her at shliu@microsoft.com.*

Furu Wei *is a staff researcher at the IBM China Research Lab. His research interests include natural-language processing, information retrieval, and text mining. Wei has a PhD from Wuhan University. Contact him at weifuru@cn.ibm.com.*

Michelle X. Zhou *is a research senior manager at IBM Research – Almaden. Her research interests include the interdisciplinary areas of intelligent user interaction, smart visual analytics, and people-centric information management. Zhou has a PhD in computer science from Columbia University. Contact her at mzhou@us.ibm.com.*

Huamin Qu *is an associate professor in the Hong Kong University of Science and Technology's Department of Computer Science and Engineering. His research interests include visualization and computer graphics. Qu has a PhD in computer science from Stony Brook University. Contact him at huamin@cse.ust.hk.*