Machines in the conversation: Detecting themes and trends in informal communication streams

W. S. Spangler
J. T. Kreulen
J. F. Newswanger

Data-mining techniques that detect trends and patterns in structured data are often ill-suited for analysis of unstructured text. Information critical to business—and generated by groups such as employees, customers, and the public—appears in such forms as chats, electronic discussion forums, and blogs. This paper describes techniques developed to detect themes and trends in such informal communication streams. Our approach begins with unsupervised text clustering to create initial categories. A human analyst then refines the categories into easily understandable themes. To facilitate this process, we developed an interactive approach to text category creation and validation that aids the analyst in evaluating each category of a taxonomy and makes it possible to visualize relationships among categories. The resulting analysis can then be communicated to participants in real time. We report on the results of using these techniques in IBM companywide "Jam" events, during which tens of thousands of employees worldwide participated in electronic discussions of key business issues.

INTRODUCTION

Since the dawn of spoken language, conversation has been the means by which ideas are developed and a consensus around those ideas obtained. The speed, range, and modes in which conversations can take place have increased with technological advancements over time. Recently, developments in the Internet and associated applications have made it possible for the scale of a single conversation to grow to one involving the simultaneous input of thousands of people. A discourse this massive poses the new challenge of properly summarizing all the thoughts generated and making them comprehensible for participants. This is the problem we address in our research.

Machines taking part in conversations is not a new idea. Conversation between man and machine has been a subject of intense interest ever since the computer was invented. The famous Turing Test[1] for machine intelligence focused on a machine being indistinguishable from a human in one-on-one conversation. One of the first artificial intelligence programs, ELIZA,[2] was a demonstration of a rudimentary conversation between a human patient and a machine "counselor". Our research takes on one small piece of the overall Turing Test problem by seeking an answer to the question, "What can computers contribute to a discourse that extends conversational content beyond what humans convey on their own?"

■ The role of the computer in the discussion can be a combination of facilitator, neutral observer, and reporter ■

We believe the answer to this question lies in the text analysis of informal electronic communication streams. A computer that is recording and observing an electronic conversation among many different individuals over a period of time may be able to detect and report on overall metalevel themes and trends in the conversation, relay this information back to the conversational group, and thereby contribute to—and even influence—the course of the conversation. The theory is that in large-scale conversations, such as those taking place on Internet forums and through blogs (Web sites used in the manner of online journals), there are bound to be *emergent phenomena*, themes and trends that reflect common aggregate behavior that no single human reader can easily discern. This is where text-mining approaches come in: The role of the computer in the discussion can be a combination of facilitator, neutral observer, and reporter—helping each human participant to more fully understand and appreciate all of the other human participants' thoughts and ideas and helping to amplify those discussion points that seem to reflect areas of group consensus or overlapping interests. Once an electronic discussion reaches a certain critical size (e.g., those involving hundreds or even thousands of participants in a focused period of time), the need for an individual or individuals to play this role becomes readily apparent. But, as the size of the conversation grows, the sheer volume of the content makes it impractical for humans to fulfill this role successfully. Thus we believe that as conversations scale larger and larger, enabled by instant messaging and World Wide Web technology, the need for computers to be involved in analyzing the content of the conversation and contributing the findings to the conversation becomes greater and greater.

The role played by computers in furthering human discussion is just beginning to be explored in research. The unstructured nature of blogging, discussion groups, opinions, reviews, and the like creates a kind of intellectual democracy of ideas.[3] Additionally, research has shown that group editors with shared concurrent editing capability have a positive effect on brainstorming.[4] Taking this a step further, it has also been shown that directed brainstorming[5] has a positive effect on creativity in problem solving. Obviously it is important to understand the organization[6,7] of the information. Then it is necessary to understand how this organization changes and what the diffusion characteristics[8] of ideas are over time. Once the behavior over time is understood, we would then want to understand the causal nature and the influential effects of information in a network.[9] For some applications, one may want to use and model this understanding to predict future behaviors.[10]

Our research is not about inventing new text-analysis tools; it is about employing and combining existing text-mining techniques in a new way to analyze and contribute to human discourse. We have developed a systematic method and toolset, which we first described in Reference 11. This paper describes how we have taken that generic text-mining approach and applied it to large-scale conversations called *Jams*.[12,13] A Jam is a construct invented at IBM that allows an organization of significant size to have a discussion in an area of interest with the goal of building consensus around actionable ideas. Our previous work began to indicate the potential of this technology to help facilitate the conversation during a single IBM Jam.[14] This paper takes a much broader look at both the methodology and its application across several Jams (internal to IBM and external) and shows how the analysis techniques have evolved to meet the challenges of this particular application. The success we have had with our approaches to date shows this to be a promising area for future applications in the field of conversational analytics and human-machine interaction.

WHAT IS A JAM?

A Jam is an internet- or intranet-based discussion and idea-stimulation vehicle. More formal than a chat room, a Jam is typically organized into a handful of separate forums (from four to seven in number), each on a different subtopic related to the overall Jam topic. The Jam is continuous, but conducted only for a limited time period (usually between 48 and 72 hours). During the event, participants can come into and leave a Jam as often as they like. Participants who register at the site can make original posts or reply to existing posts. The posts are labeled with the participant's name (anonymous contributions are not permitted). Some Jam participants may simply read the existing posts while others will enter posts without reading anyone else's thoughts. Most participants will both read what is already in the Jam and make their own contributions. As the Jam continues, themes emerge from the communication stream. These themes, detected by text mining, are posted back to the Jam periodically along with typical comments for each theme. This allows participants to see at a glance the gist of what is being said.

Moderators in each forum can highlight hot topics, referred to as *Jam Alerts*, as they emerge in the discussion (this is separate from the themes detected by text mining). Participants can also use full text search to browse for posts on a certain subject or for posts that particular individuals have contributed. Finally, posts can be e-mailed by Jam participants to others, perhaps encouraging them to make new contributions.

The process of Jamming at IBM has evolved over several years. At first it involved no text-mining technology at all. It used only human facilitators and asked participants to rate ideas to help analyze the event as it was happening and communicate information back to participants. Unfortunately, this system suffered an inevitable problem: The early ideas usually got the most votes. With the introduction of text-mining techniques into the more recent Jam events, each individual participant in the Jam is provided with the necessary information to "hear" the Jam as a whole.

At this writing, there have been seven Jams sponsored by IBM. This paper focuses on the three most recent Jam events that took place at different times between August 2003 and December 2005. *ValuesJam*, a 72-hour event in 2003, involved IBM employees and explored the company's fundamental business beliefs and values. *WorldJam*, held the following year within IBM, studied how the IBM Values could be implemented. This 48-hour event generated over 32,000 posts. *HabitatJam*, sponsored by the United Nations Habitat Initiative, the government of Canada, and IBM in 2005, was an open discussion on the Internet about the future of cities and the search for solutions to critical worldwide urban issues. During this 72-hour event, over 15,000 posts were generated from participants in 120 different countries.

UNDERSTANDING THE JAM THROUGH INTERACTIVE TEXT MINING

Although computers are quite capable of grouping documents together based on their surface characteristics (word frequency), such groupings may not always be useful. To ensure that categories make sense and make useful distinctions can require common sense knowledge and reasoning of a type not yet exhibited reliably by computer software. The involvement of a human in the role of analyst is needed to identify and discard spurious classes that are created from common features but have no underlying semantic value. This is what we mean by interactive text mining.

To play this critical role, the human data analyst must be provided with the necessary information to understand the meaning of each class. When one considers that each class may be composed of hundreds of examples and that the data frequently needs to be analyzed for multiple forums in real time, it becomes clear that powerful summarization tools are needed to communicate the meaning of each class in the taxonomy to the data analyst. Furthermore, as the data analyst finds classes that need to be modified or removed from the taxonomy, powerful editing tools are required to make changes that reflect the analyst's intent.[11,14]

Generating a taxonomy

The initial taxonomy is an important first step in helping the human analyst make sense of a large set of documents quickly and accurately. Our methodology provides two main alternatives for taxonomy generation: *K-means* clustering (using a set of randomly selected k centroids—average term vectors—to generate clusters) and *cohesive keyword* clustering (generating clusters based on specific words or phrases selected on the basis of a cohesion metric). We employed the K-means clustering method for the two IBM Jams and the cohesive keyword method for HabitatJam.

Taxonomy generation through clustering

In cases where the user has no preconceived idea about what categories the document collection should contain, text clustering may be used to create an initial breakdown of the documents into clusters, grouping together documents having similar word content.

■ More formal than a chat room, a Jam is typically organized into a handful of separate forums, each on a different subtopic related to the overall Jam topic ■

To facilitate this process we represent the documents in a vector space model. We represent each document as a vector of weighted frequencies of the document features (words and phrases).[15] We use the txn weighting scheme, also known as normalized term frequency.[16] This scheme emphasizes words with high frequency in a document and normalizes each document vector to have a unit Euclidean norm, i.e., the magnitude of each feature vector is 1.0. For example, if a document were simply the sentence, “We have no bananas, we have no bananas today,” and the dictionary consisted of only two terms, “bananas” and “today”, then the unnormalized document vector would be {21} (to indicate two “bananas” and one “today”), and the normalized version would be

$$\left[2/\sqrt{5}, 1/\sqrt{5}\right].$$

The words and phrases that make up the document feature space are determined by first counting which words occur most frequently in the text (in the most documents). A standard *stop-word list* is used to eliminate words such as “and”, “but”, and “the”.[17] The top N words are retained in the first pass, where the value of N may vary depending on the length of the documents, the number of documents, and the number of categories to be created. Typically $N =$ 2,000 is sufficient for 10,000 short documents of about 200 words to be divided into 30 categories. (Note that 30 categories were chosen based on user feedback concerning how many categories they could readily contemplate during analysis.) After selecting words in the first pass, we make a second pass to count the frequency of phrases that occur using these words. A phrase is considered to be two consecutive words occurring without intervening nonstop words. We again prune to keep only the N most frequent words and phrases. This becomes the feature space. The documents are then indexed by their feature occurrences (i.e., word count) in a third pass through the data. The user may edit this feature space as desired to improve clustering performance. For instance, the user can add particular words and phrases deemed important, such as named entities like “International Business Machines”. Stemming (reducing words to their roots so that different forms of the same word are selected) is usually incorporated to create a default synonym table that the user may edit.[18]

For categorization, we employ the K-means algorithm,[19,20] using a cosine similarity metric[21] to partition the documents into k disjoint clusters automatically. The algorithm is very fast and easy to implement. See Reference 21 for a detailed discussion of various other text-clustering algorithms. The K-means algorithm produces a set of disjoint clusters and a centroid for each cluster that represents the cluster mean. Typically k is initially set to 30 for the highest level of the taxonomy, though the user may adjust this if desired. The initial taxonomy assigns each document to only one category (cluster). After clustering is complete, a final merging step takes place. In this step, two or more clusters dominated by the same keyword (dominated means that 90 percent of the examples contain this keyword) are merged into a single cluster, and a new centroid is calculated based on the combined example set. We do this to avoid arbitrarily separating similar examples into different subsets before the analyst evaluates the class as a whole.

To help the analyst understand the meaning of each cluster, the system names each document category. Cluster naming is not an exact science, but our method attempts to describe the cluster as succinctly as possible without missing any important constituent components. The first rule of naming is that if a single term dominates a cluster, then this term is given as the cluster name. If no term dominates, then the most frequent term in the cluster becomes the first word in the name and the remaining set of examples (those not containing the most frequent term) are analyzed to find the dominant term. If a dominant term for the remaining examples is found, then this term is added to the name (separated by a comma), and the name is complete; otherwise, the process continues for up to four words. Beyond four words, we simply call the class “Miscellaneous”.

Taxonomy generation through cohesive terms

During early Jams in which we used our text-mining approach, one of the feedback comments we received was that the initial categorization was often difficult to interpret, making the process of refining the categories painfully slow. It turns out that one of the drawbacks of the K-means clustering approach is that it frequently creates categories which are difficult to interpret by a human being. Approaches to cluster naming attempt to address this issue by adding more and more terms to a name to capture the complex concept that is being modeled by a centroid. An example from our own ValuesJam of a difficult cluster name would be: *world, specific, develop, e-business*. Unfortunately, this approach puts the onus on the human interpreter to make sense of what the list of words means and how it relates to the entire set of examples contained in the category.

To address this problem and speed the taxonomy editing process by starting with category names that are easier to comprehend, we developed a new strategy (described here for the first time) for document categorization based on categories centered around selected individual terms in the dictionary. We then employ a single iteration of K-means to the generated categories to refine the membership so that documents which contain more than one of the selected terms can be placed in the category best suited to the overall term content of the document. Note that the alternative strategy of putting such documents in more than one category (i.e., multiple membership) is less desirable because it increases the average size of each category and defeats the purpose of summarization by the divide-and-conquer strategy inherent in document clustering. Creating multiple copies of documents that match more than one category would be multiplying instead of dividing. Once the clusters are created, we name each one, using the single term that created it, thus avoiding the complex name problem associated with K-means clusters.

Selecting which terms to use for generating categories is critically important. Our approach is to rank all discovered terms in the data set based on a normalized measure of cohesion calculated using

$$cohesion(T, n) = \frac{\sum_{x \in T} \cos(centroid(T), x)}{|T|^n},$$

where T is the set of documents that contain a given term, $centroid(T)$ is the average vector of all these documents, and n is a parameter used to adjust for variance in category size (typically $n = 0.9$). The cosine distance between document vectors is defined to be

$$\cos(X, Y) = \frac{X \cdot Y}{||X|| \cdot ||Y||}.$$

Terms that score relatively high with this measure tend to be those with a significant number of examples having many words in common. Adjusting the n parameter downward tends to surface more general terms with larger matching sets, while adjusting it upward gives more specific terms.

The algorithm selects enough of the most cohesive terms to get 80 to 90 percent of the data categorized. Terms are selected in cohesive order, skipping those terms in the list that do not add a significant number (e.g., more than three) of additional examples to those already categorized with previous terms. The algorithm halts when at least 80 percent of the data has been categorized and the uncategorized examples are placed in a Miscellaneous category. The resulting categories are then refined using a single iteration of K-means (i.e., each document is placed in the category of the nearest centroid as calculated by the term membership just described).

While this approach does not completely eliminate the need for taxonomy visualization and editing by an analyst (as described in the following sections), it does make the process much less cumbersome by creating categories that are, for the most part, fairly easy to comprehend immediately. In practice, this cut the time required to edit each taxonomy by about half (from around 30 minutes to around 15 minutes per forum in the Jam).

Viewing the taxonomy

Before analysts can begin editing a taxonomy, they must first understand the existing categories and their relationships. In this section, we describe our strategy to communicate the salient characteristics of a document taxonomy to the user.

Our primary representation of each category is the centroid.[19] The distance metric employed to compare documents to each other and to the category centroids is the cosine similarity metric.[21] As we describe later in the section "Editing the taxonomy," we are not rigid in requiring that each document belong to the category of its nearest centroid, nor do we strictly require every document to belong to only one category.

■ Our research is not about inventing new text-analysis tools; it is about employing and combining existing text-mining techniques in a new way to analyze and contribute to human discourse ■

Summaries

Because we cannot expect the analyst to have time to read through all of the individual documents in a category, summarization is an important tool to help the user understand what a category contains. Summarization techniques based on extracting text from the individual documents[22] were found to be insufficient in practice for the purpose of summarizing an entire document category, especially when the theme of that category covered diverse elements. Instead, we employ two different techniques to summarize a category. The first is a feature bar chart. This chart has an entry for every dictionary term (feature) that occurs in any document of the category. Each entry consists of two bars, a red bar to indicate the percentage of the documents in the category that contain the feature and a blue bar to indicate how frequently the feature occurs in the background population of documents from which the category was drawn. The bars are sorted in decreasing order of the difference between blue and red. Thus the most important features of a category are shown at the beginning of the chart with their relative importance indicated by the size of the bars.

The second technique is a dynamic decision tree representation that describes the feature combinations that define the category. This tree is generated in the same manner as a binary ID3,[23,24] selecting at each decision point the attribute that is most helpful in splitting the document universe so that the two new classes created are most nearly pure category and pure noncategory. Each feature choice is made dynamically as the user expands each node until a state of purity is reached or when no additional features will improve the purity. The result is essentially a set of classification rules that define the category to the desired level of detail. At any point the user may select a node of the decision tree to see either all the documents at the node, all the in-category documents at the node, or all the non-category documents at the node. The nodes are also color coded: red for a node whose membership is 50 percent or more in-category and blue for a node whose membership is less than 50 percent in-category. This display gives users an in-depth definition of the class in terms of salient features and lets the analyst readily select various category components for further study.

Visualization

We employ a visualization strategy to understand how two or more categories at the same level of the taxonomy relate to each other. The idea is to visually display the term vector space model for each document so that the documents will appear as points in space. The result is that documents containing similar words occur near each other in the visual display. If the vector space model were two dimensional, this would be straightforward: We could simply draw the documents as points on an X,Y scatter plot. The difficulty is that the document vector space is of much higher dimension. In fact, the dimensionality is the size of the feature space (dictionary), which is typically thousands of terms. Therefore, we need a way to reduce the dimensionality from thousands to two in such a way as to retain most of the relevant information. Our approach uses the CViz method,[25] which relies on three category centroids to define the plane of most interest and to project the documents as points on this plane (by finding the intersection with a normal line drawn from point to plane). The selection of which categories to display in addition to the selected category is based on finding the categories with the nearest centroid distance to the selected category. The documents displayed in such a plot are color coded according to category membership. The centroid of the category is also displayed. An example of the resultant plot is shown in *Figure 1*. Such a plot is a valuable way to discover relationships among neighboring concepts in a taxonomy. For instance, it might reveal overlaps that require further investigation.

Sorting examples

When a user wants to study the examples in a category to understand the essence of the category, it is important that the examples not be chosen at random. A random selection can sometimes lead to a skewed understanding of the category content, especially if the sample is small compared with the size of the category (often the case in practice). To overcome this potential problem, our software enables examples to be sorted based on the criteria of *most typical* first or *least typical* first. This translates in vector space terms to sorting in order of distance from the category centroid (i.e., the most typical example is closest to the centroid, the least typical example is farthest from the centroid). The advantage of sorting in this way is twofold. Reading documents in the most typical order can help the user quickly understand what the category is generally about without having to read a large sample of documents in the category; reading the least typical documents can help the user understand the scope of the category and determine if there is conceptual purity.

Editing the taxonomy

Once the analyst understands the meaning of the classes in the taxonomy and their relationship with one another, the next step is to provide tools for rapidly changing the taxonomy to reflect the needs of the application. Our goal here is not to produce a perfect taxonomy for every point of view because such a taxonomy may not exist or may require too much effort to obtain. Instead we want to focus the user's efforts on creating a natural taxonomy that can summarize major themes in the discussion, thus eliminating any categories that the system may have created that do not make sense as discussion themes. This might be due to a centroid forming around a concept that is syntactically similar but has different meanings in different contexts. For example, a cluster created around the word “customer” might be based on two types of comments: one set dealing with customer relationship management applications and another set dealing with customer satisfaction issues. In some cases such changes can be made at the category level; in other cases a more detailed modification of category membership may be required. Our tool provides capabilities at every level of a taxonomy to allow the user to make the desired modifications with a simple point and click.

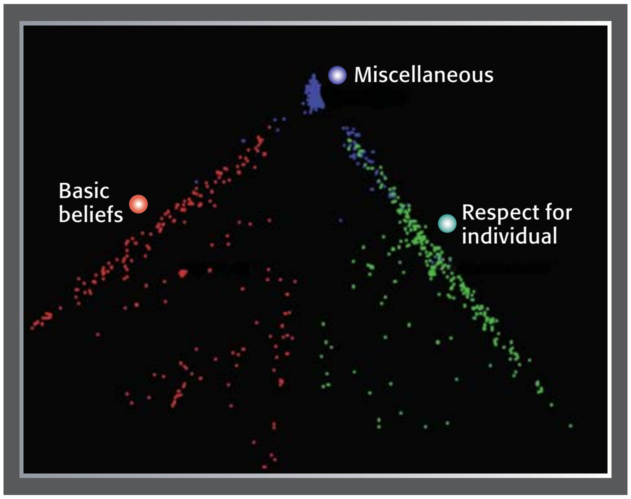


Figure 1
Class visualization

Category-level changes

Category-level changes involve modifying the taxonomy at a macro level without direct reference to individual documents within each category. Categories can be merged or deleted.

Merging two classes means creating a new category that is the union of two or more previously existing category memberships. A new centroid is created that is the average of the combined examples. The user gives an appropriate name to the new category.

Deleting a category (or categories) means removing the category and its children from the taxonomy. This, however, may have unintended consequences because all the examples that formerly belonged to the deleted category must now be placed in a different category at the current level of the taxonomy. To make this decision more explicit for the user, we introduced a pie chart that shows all of the secondary classes and the percent of the category's documents that would be assigned to each if the category were to be deleted. Each slice of the pie chart can be selected to view the individual documents represented by the slice. Making this information explicit allows the user, when faced with a decision concerning the deletion of a category, to arrive at an informed decision and avoid unintended consequences.

Document-level-changes

While some changes to a taxonomy may be made at the class level, others require a finer degree of control. These are called document-level changes and consist of moving or copying selected documents from a source category to a destination category. The difficult part of this operation from the user's point of view is selecting the right set of documents to move so that the source and destination categories are changed in the manner desired. To address this problem, three methods of selection are provided.

1. *Selection by keywords*—One of the most natural and common ways to select a set of documents is with a keyword query. The user may enter a query for the whole document collection or for a specific category. The query can contain keywords and use Boolean logic. Words that co-occur with the query string are displayed to help the user refine the query. Documents that are found by using the keyword query tool can be viewed immediately and selected one at a time or as a group to move or establish a new category.
2. *Selection by sorting*—Another way to select documents to move or copy is by the most typical or least typical sorting technique described earlier. For example, the documents that are least typical of a given category can be located, selected, and moved or placed in a new category.
3. *Selection by visualization*—The scatter plot visualization display (Figure 1) can also be a powerful tool for selecting individual or groups of documents. Groups of contiguous points (documents) can be selected by using the mouse to draw a floating box around them, and then they can be moved to a new category.

Validation

Whenever a change is made to the taxonomy, it is very important for the analyst to validate that the change has had the desired effect on the taxonomy as a whole and that no undesired consequences have resulted from unintentional side effects.[26] Our software contains a number of capabilities that allow the user to inspect the results of modifications. The goal is to ensure that all the categories are meaningful, complete, and differentiable, and that the concepts represented by the document partitioning can be carried forward automatically in the future as new documents arrive.

Direct inspection

The simplest method for validating the taxonomy is through direct inspection of the categories. The category views described earlier in the subsection "Viewing the taxonomy" provide unique tools for validating that the membership of a category is not more or less than what the category means. Looking over some of the least typical documents is a valuable way to ascertain quickly that a category does not contain documents that do not belong. Another visual inspection method is to look at the nearest neighbors of the category being evaluated through the scatter plot display. Areas of document overlap at the margins are primary candidates for further investigation and validation.

Validation metrics

Much research has been done in the area of evaluating the results of clustering algorithms.[17,27] While such measures are not entirely applicable to taxonomies that have been modified to incorporate domain knowledge, there are some important concepts that can be applied from this research. Our vector space model representation[15,16] (admittedly a coarse reflection of the documents' actual content) at least allows us to summarize a single level of the taxonomy with some useful statistics, including cohesion and distinctness. *Cohesion* is a measure of similarity within a category. This is the average cosine distance of the documents within a category to the centroid of that category. *Distinctness* is a measure of differentiation between categories. This is one minus the cosine distance of the category to the centroid of the nearest neighboring category.

These two criteria are variations on the ones proposed by Berry and Linhof: compactness and separation.[28] The advantage of using this approach as opposed to other statistical validation techniques is that these criteria are more easily computed and also readily understood by the taxonomy expert. In practice, these metrics often prove useful in identifying two potential areas of concern in a taxonomy. The first potential problem is having Miscellaneous classes. These are classes that have a diffuse population of documents with widely varying contents. Such classes may need to be split further or subcategorized. The second potential problem is when two different categories have very similar content. If two or more classes are almost indistinguishable in terms of their word content, they may be candidates for merging.

Statistical measures such as cohesion and distinctness provide a good rough measure of how well the word content of a category reflects its underlying meaning. For example, if a user-created category is not cohesive, then there is some doubt as to whether an analyst could learn to recognize a new document as belonging to that category as the word content is not well-defined. On the other hand, if a category is not distinct, then there is at least one other category containing documents with a similar vocabulary. This means that an analyst may have difficulty distinguishing into which of the two similar categories to place a candidate document. Of course, cohesion and distinctness are rough and relative metrics, so there is no fixed threshold value at which we can say that a category is not cohesive enough or lacks sufficient distinctness. In general, whenever a new category is created, we suggest that the cohesion and distinctness scores for the new category be no worse than the average for the current level of the taxonomy.

Emerging themes

In addition to overall themes for each forum, it is also desirable to discover newly emerging issues in the discussion. One way to discover such themes would be to generate a new taxonomy for each of the forums based on only the most recent sample of the data.

There are several drawbacks to this approach, not the least of which is that categories generated in this way may differ from the overall categories for reasons that are not related to the different data sample, but are inherent artifacts of the clustering approach, that is, the fact that K-means clustering begins with a random starting point. A simpler, more reliable way to find emerging themes is to analyze the dictionary of terms across time to determine which terms are showing increased mentions. To achieve a reliable sample size, we defined *recent* to be the last 10 percent of the posts in a forum, sorted chronologically. We then analyzed all the dictionary terms to determine which, if any, occurred with an unusually high frequency in the Recent set. *Unusually high* is determined by using a chi-squared test, which determines the independence of two discrete random variables.[29] Terms that occur with probability of less than 0.01 are selected. The resulting term list is displayed to the user for further investigation by trend charts and example displays that can be used to create new document categories, which can then be published as themes.

■ Inevitably, computers are becoming a greater and greater participant in our conversations ■

CASE STUDIES

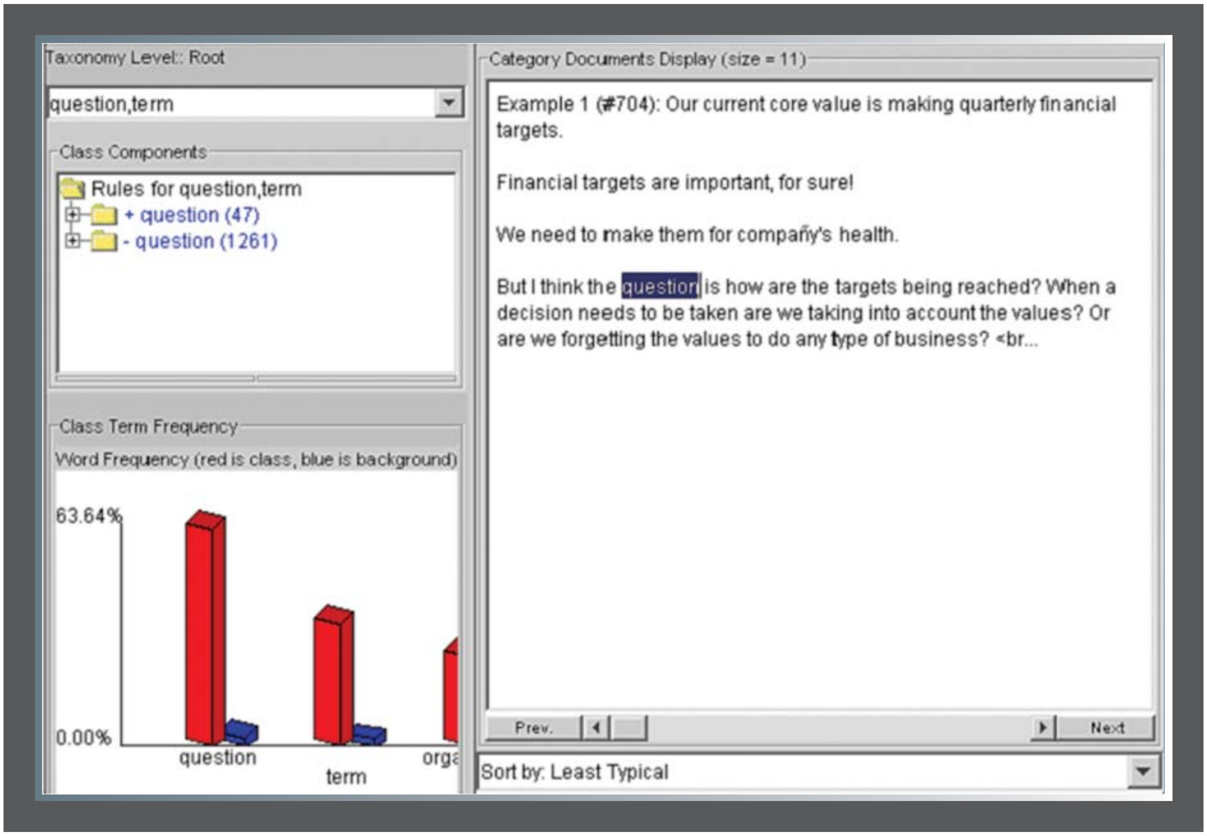
We are focusing on three major Jams that recently took place, two internal for IBM employees worldwide and one for the World Urban Forum. In this section, we describe how text analysis has been used in each of these Jams and how it has evolved to meet the demands of this new medium.

ValuesJam

ValuesJam was a 72-hour global brainstorming event on the IBM intranet, held July 29–August 1, 2003. IBMers described their experiences and contributed ideas by means of four asynchronous discussion forums. The purpose of real-time interactive text mining of the Jam was to generate forum topics that allowed participants to identify themes as they emerged in each forum and in the Jam overall, in 12-hour intervals. Total posts for this event were in excess of 8,000 over the course of the event, with one of the largest forums containing more than 3,000 posts.

Analyzing discussion forum data to produce topic areas of interest presents several challenges that an interactive text-mining approach is well-suited to address:

1. The forum analyzer must produce categories that reflect meaningful groups of posts, and these groups must not contain a significant number of extraneous or misclassified examples.
2. Each cluster of posts must be given a concise yet meaningful name.
3. When a cluster of posts is presented, a set of representative examples are needed to further explain the meaning of the cluster and direct the user to the appropriate point in the discussion.



Taxonomy Level:: Root

question,term

Example 1 (#704): Our current core value is making quarterly financial targets.

Financial targets are important, for sure!

We need to make them for compañy's health.

But I think the question is how are the targets being reached? When a decision needs to be taken are we taking into account the values? Or are we forgetting the values to do any type of business? <br...

Sort by: Least Typical

Figure 2
Text-clustering class view

4. The clusters need to evolve with the discussion, adding new clusters over time as appropriate to incorporate the new topics that arise without losing the old clusters and thus the overall continuity of the discussion topic list.

Clearly a completely automated solution is impractical, given these requirements, and a manual approach requiring a set of human editors to read over 8,000 posts in 72 hours and classify them is prohibitively expensive. Interactive text mining is thus an ideal candidate for this application. During ValuesJam, different experts in each forum used our tools to develop themes for that forum, and a single primary analyst (one of the authors of this paper) helped coordinate the analysis as a whole.

Initial taxonomy generation

The first taxonomy generated for discussions in the largest ValuesJam forum was created on 1,308 posts representing 20 hours of discussion. The form of the taxonomy was a list of 24 classes that indicated their name, size, cohesion, and distinctness.

We began by sorting the categories by their cohesion scores. This gave us a useful order in which to tackle the problem of quickly understanding the taxonomy, category by category. We viewed each category in detail, made adjustments as necessary, and gave the category a new name if needed (e.g., the category name "stock price" replaced the name "stock" given by the system). Occasionally we found clusters that were formed based on words that were not relevant to the content of the post, such as the "question,term" cluster in ***Figure 2***. For this class, we viewed the secondary class pie chart to determine where the examples would go when the centroid was removed. We saw that they would be distributed evenly throughout the taxonomy, so we felt we could delete the centroid without ill effect.

The Miscellaneous class required special attention. Individual dictionary terms can frequently be used to extract a common set of examples from a Miscellaneous class and create a useful separate category. In *Figure 3*, the category centered on the word “trust” is an example. Clicking on the red *trust* bar in the figure caused all examples in Miscellaneous that contained the word “trust” to be selected. These were then further edited, and a new category called “trust” was created in the taxonomy. Finally, the complete analyst-adjusted list of categories was generated.

Using our methodology and software text-analysis tools, this entire process required about a half hour of concentrated effort. We then used this information to generate reports to the ValuesJam audience. The resulting Web page report is shown in *Figure 4*. Selecting any of the links shown in the figure took the user to a display of ten of the most typical comments for that theme. This process was then repeated for each of the remaining forums and for the Jam as a whole. The entire reporting operation took about three to four hours.

ValuesJam emerging themes

As the Jam progressed, new topics naturally emerged. To identify these, the emerging themes analysis described earlier was especially valuable. A good example of this came late in the Jam when a breaking news story had an impact on the discussion.[30] We observed the word “pension” occurred 51 times overall, and 11 times in the last 10 percent of the data. This was deemed by the chi squared test to be a low probability event ($P = 0.0056$). The trend line for this keyword, shown in *Figure 5*, indicates the spike. Posts that contained the word “pension” had been decreasing as a percent of total posts, but on the last day there was a sharp increase. Looking at the text for these examples quickly revealed the cause—the breaking news story—and thus a new category was created centered on this word.

Success of interactive text mining during ValuesJam

Our interactive text-mining approach, with only one primary analyst working on a standard laptop, showed itself to be very capable of supporting real-time analysis of a discussion among thousands of users. A survey that included 1,248 respondents done after ValuesJam indicated that 42 percent of the participants used the theme pages to enter the Jam. Of those who used this feature, 72 percent found it to be important and 61 percent found it to be satisfactory—the top two possible ratings. Only 10 percent were dissatisfied.

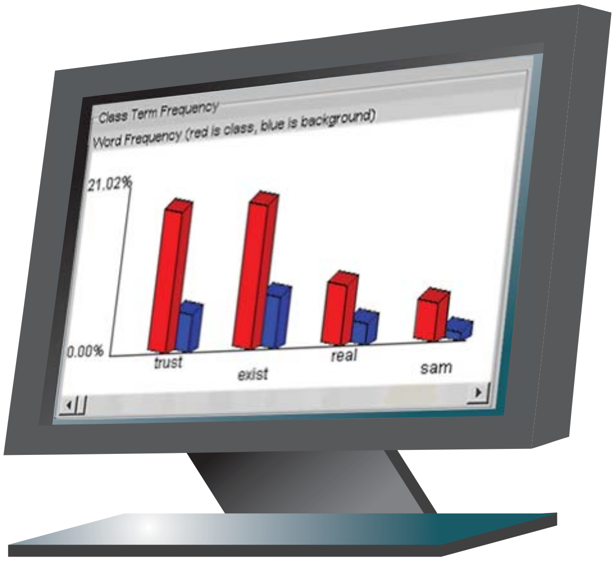
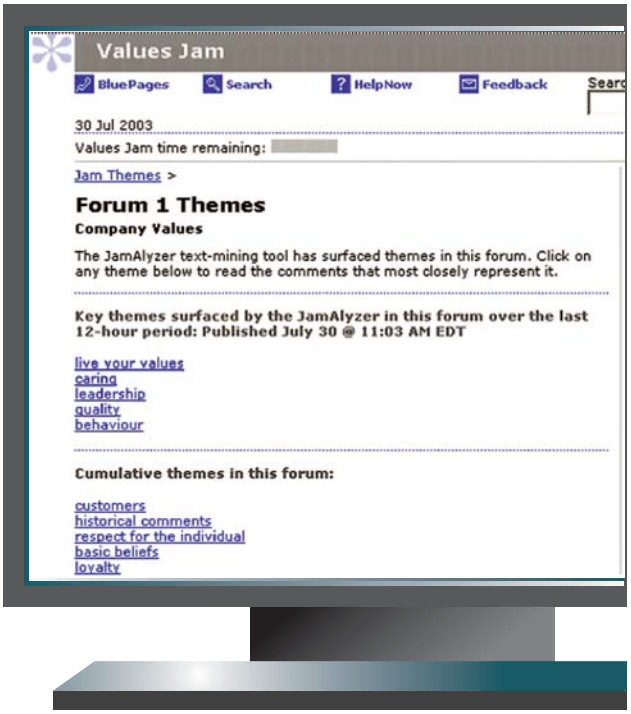


Figure 3
Miscellaneous class



30 Jul 2003

Values Jam time remaining:

Jam Themes >

Forum 1 Themes

Company Values

The JamAlyzer text-mining tool has surfaced themes in this forum. Click on any theme below to read the comments that most closely represent it.

Key themes surfaced by the JamAlyzer in this forum over the last 12-hour period: Published July 30 @ 11:03 AM EDT

live your values
caring
leadership
quality
behaviour

Cumulative themes in this forum:

customers
historical comments
respect for the individual
basic beliefs
loyalty

Figure 4
Jam Theme Web page presented to participants

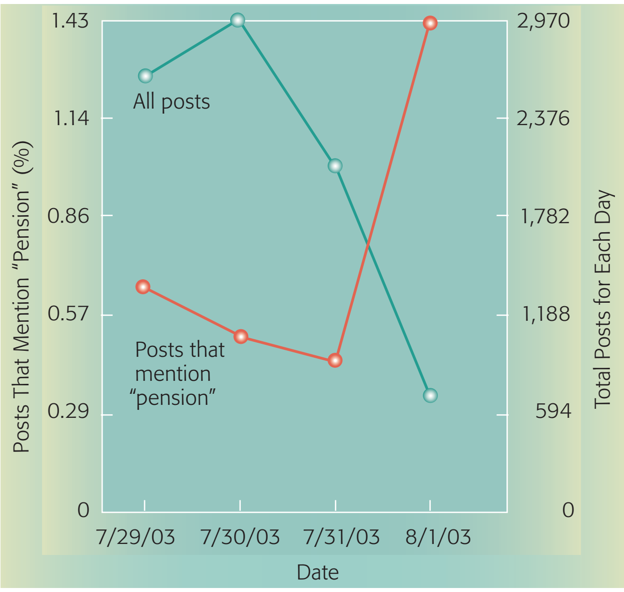


Figure 5
Trend line for the word "pension" over time during ValuesJam

IBM WorldJam

The purpose of IBM WorldJam, a companywide Jam held in 2004, was to encourage ideas about how IBM could best implement its values. The process we used to generate Jam Themes was much the same as in ValuesJam. After the Jam, a survey was conducted to determine the success of the event and the usefulness of the various tools involved. One of these tools was text analysis as presented by Jam Theme pages. As the data in *Table 1* shows, only the WorldJam2004 search tool came close to Jam Themes when rated for importance. The frequency of use and satisfaction scores for Jam Themes surpassed the other Jam discovery tools.

World Urban Forum HabitatJam

In HabitatJam, we exposed the Jam technical infrastructure to a non-IBM audience for the first time. The purpose of the Jam was to identify topics for discussion at an upcoming World Urban Forum. This was a 72-hour event that included participants from 120 countries. Posts were submitted in both French and English in seven forums.

During the event, text analysis was done three times a day for each of the seven forums and three times a day for the Jam as a whole. The English forums were all done by a single human analyst using the techniques described in this paper (including clustering using cohesive terms). Despite the fact that nearly twice as many themes were generated in the same time period for this Jam as for WorldJam, the post-event survey indicates that the quality (as

Table 1 Post-Jam survey results

(Randomized list) Values in cells are means	Frequency of use 3 = Used often 2 = Used a few times 1 = Used once 0 = Did not use	Importance 5 = Very important 4 = Somewhat important 3 = Neither important nor unimportant 2 = Somewhat unimportant 1 = Very unimportant	Satisfaction 5 = Very satisfied 4 = Somewhat satisfied 3 = Neither satisfied nor dissatisfied 2 = Somewhat dissatisfied 1 = Very dissatisfied
WorldJam 2004			
Search tool[†]	0.8	4.1	2.8
Jam Alert[††]	0.8	3.5	3.4
Jam Themes[†††]	1.8	4.2	3.8
Web mail[††††]	0.2	3.4	3.1
HabitatJam 2004			
Search tool	1.6	4.1	3.4
Jam Alert	1.0	3.4	3.3
Jam Themes	2.1	4.2	3.8
Web mail	0.6	3.4	3.0

[†] The search box in the upper right corner of Jam pages
[††] Notices posted on the Jam home page and Forum pages; usually news or featured links
[†††] Pages on the Jam site where central themes are displayed overall and by Forum
[††††] The ability to send or receive Jam posts as e-mail to or from other participants

measured by importance and satisfaction scores) did not degrade. This survey was conducted following the Jam ($n = 1{,}374$ respondents). The results, presented in the lower section of Table 1, indicate that for this event, the Jam Themes Web page was both the most important and the most satisfactory discovery tool available during the event.

The user feedback scores from all three Jams, with different audiences and topic areas, show that Jam Themes are the most frequently used tool for navigation and discovery—even more so than text search. As well, the overall satisfaction was higher for Jam Themes than for any other Jam tool. While there is still room for improvement, these results indicate that our text-mining approach has significant value for discussion participants.

SUMMARY AND FUTURE WORK

We have demonstrated the value of using text-mining techniques to facilitate and enhance large-scale electronic dialogs. While it is true that the computer does not take part directly in the discussion in the same way as a human participant would, it still, in conjunction with the human analyst, plays a critical role in generating content that furthers the discussion. In fact, the computer plays a role in Jam conversations that might be played by a human being in a much smaller conversation—that of facilitator or moderator—by helping to ensure that all points of view are heard and taken into account by all participants.

A problem that still needs to be addressed is that those themes that become established as the Jam progresses may tend to become document silos, ignoring potential relationships among and between other themes. We are experimenting with semantic browsing approaches that might help alleviate this problem for users in the future.

Many of the techniques described in this paper can be applied in fields beyond Jams. Market intelligence, through which businesses seek to find actionable insights for economic advantage, is a common application.[31] This can also be accomplished by using alternative information sources such as captured information from customer interactions during sales and service. Additionally, in large enterprises there is abundant text information available in e-mails, documents, and databases that can be leveraged in a similar way.

The planned future direction of our work is to minimize the need for a human analyst or perhaps ultimately eliminate it, leaving the computer alone to play the role of Jam Theme generator and conversation facilitator. This will require more precise text category naming strategies and intelligent pruning techniques for removing categories that are not meaningful or helpful in summarizing a topic area. Perhaps the conversation participants themselves might be enlisted to provide feedback on categories that might be used to adjust the text categorization algorithms.

Inevitably, computers are becoming a greater and greater participant in our conversations. Through text analysis, they can tell us things that humans would find difficult or even impossible to discover on their own about what is being said. As text-analysis techniques become ever more powerful and intuitive, the role of machines in our conversations is only going to increase in the future. We look forward eagerly to hearing what they have to say.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contributions of Mike Wing for inventing Jams in the first place and for recognizing the important role that text mining would have to play, of Kristine Lawas, who assisted in editing themes during the Jams, and Dharmendra Modha for his text visualization and analysis insights. Finally the authors thank the IBM Jam infrastructure team for helping to make the technical details run so smoothly during each Jam event.

CITED REFERENCES

1. A. M. Turing, “Computing Machinery and Intelligence,” *Mind* **59**, No. 236, 433–460 (1950).
2. J. Weizenbaum, “ELIZA—A Computer Program for the Study of Natural Language Communication Between Man and Machine,” *Communications of the ACM* **9**, No. 1, 36–45 (1966).
3. C. R. Sunstein, “Democracy and Filtering,” *Communications of the ACM* **47**, No. 12, 57–59 (2004).
4. C. M. Hymes and G. M. Olson, “Unblocking Brainstorming Through the Use of a Simple Group Editor,” *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, Toronto, Ontario, Canada (1992), pp. 99–106.
5. E. L. Santanen, R. O. Briggs, and G.-J. de Vreede, “A Cognitive Network Model of Creativity: A Renewed Focus on Brainstorming Methodology,” *Proceedings of the 20th International Conference on Information Systems*, Charlotte, NC (1999), pp. 489–494.

6. J. Kleinberg, “Bursty and Hierarchical Structure in Streams,” *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Alberta, Canada (2002), pp. 91–101.

7. R. Kumar, J. Novak, P. Raghavan, and A. Tomkins, “Structure and Evolution of Blogspace,” *Communications of the ACM* **47**, No. 12, 35–39 (2004).

8. D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, “Information Diffusion Through Blogspace,” *Proceedings of the 13th International Conference on World Wide Web*, New York, NY (2004), pp. 491–501.

9. D. Kempe, J. Kleinberg, and E. Tardos, “Maximizing the Spread of Influence through a Social Network,” *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC (2003), pp. 137–146.

10. D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins, “The Predictive Power of Online Chatter,” *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, Chicago, IL (2005), pp. 78–87.

11. S. Spangler and J. Kreulen, “Interactive Methods for Taxonomy Editing and Validation,” *Proceedings of the 11th International Conference on Information and Knowledge Mining*, McLean, VA (2002), pp. 665–668.

12. P. Hemp and T. A. Stewart, “Leading Change When Business is Good: An Interview with Samuel J. Palmisano,” *Harvard Business Review* **82**, No. 12, 60–70 (December 2004).

13. K. Dave, M. Wattenberg, and M. Muller, “Flash Forums and ForumReader: Navigating a New Kind of Large-Scale Online Discussion,” *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, Chicago, IL (2004), pp. 232–241.

14. S. Spangler and J. Kreulen, “Interactive Methods of Taxonomy Editing and Validation,” *Next Generation of Data Mining Applications*, M. Kantardzic and J. Zurada, Editors, Wiley-IEEE Press, Piscataway, NJ (2005), pp. 495–524.

15. G. Salton and M. J. McGill, *Introduction to Modern Retrieval*, McGraw-Hill Book Company, New York (1983).

16. G. Salton and C. Buckley, “Term Weighting Approaches in Automatic Text Retrieval,” *Information Processing and Management* **4**, No. 5, 512–523 (1988).

17. C. Fox, “Lexical Analysis and Stoplists,” *Information Retrieval: Data Structures and Algorithms*,W. B. Frakes and R. Baeza-Yates, Editors, Prentice-Hall, Englewood Cliffs, NJ (1992), pp. 102–130.

18. A. Honrado, R. Leon, R. O’Donnel, and D. Sinclair, “A Word Stemming Algorithm for the Spanish Language,” *Proceedings of the 7th International Symposium on String Processing and Information Retrieval*, Curuna, Spain (2000), pp. 139–145.

19. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York (1973).

20. J. A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, New York (1975).

21. E. Rasmussen, “Clustering Algorithms,” *Information Retrieval: Data Structures and Algorithms*,W. B. Frakes and R. Baeza-Yates, Editors, Prentice-Hall, Englewood Cliffs, NJ (1992), pp. 419–442.

22. H. Jing, R. Barzilay, K. McKeown, and M. Elhadad, “Summarization Evaluation Methods: Experiments and Analysis,” *Working Notes of the AAAI Spring Symposium on Intelligent Text Summarization*, Stanford, CA (1998), pp. 60–68.

23. J. R. Quinlan, “Induction of Decision Trees,” *Machine Learning* **1**, No. 1, 81–106 (1986).

24. C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, MA (1999).

25. I. S. Dhillon, D. S. Modha, and W. S. Spangler, “Visualizing Class Structure of Multidimensional Data,” *Proceedings of the 30th Symposium on the Interface: Computing Science and Statistics*, Minneapolis, MN (1998), pp. 488–493.

26. B. Dom, *An Information-Theoretic External Cluster-Validity Measure*, IBM Research Report RJ-10219, IBM Almaden Research Center, San Jose, CA 95120 (2001).

27. M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “On Clustering Validation Techniques,” *Journal of Intelligent Information Systems* **17**, No. 2/3, 107–145 (2001).

28. M. J. A. Berry and G. Linoff, *Data Mining Techniques: For Marketing, Sales, and Customer Support*, John Wiley & Sons, Inc., New York (1996).

29. W. H. Press, B. Flannery, S. A. Teukolsky, and W. Vetterling, *Numerical Recipes in C*, Second Edition, Cambridge University Press, New York (1992), pp. 620–623.

30. M. W. Walsh, “Judge Says IBM Pension Shift Illegally Harmed Older Workers,” *New York Times*, August 1, 2003, http://query.nytimes.com/gst/abstract.html?res=F00B14F73E5A0C728CDDA10894DB404482.

31. N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo, “Deriving Marketing Intelligence from Online Discussion,” *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, IL (2005), pp. 419–428.

Accepted for publication March 27, 2006.
Published online September 29, 2006.

W. Scott Spangler
IBM Research Division, Almaden Research Center,
650 Harry Road, San Jose, California 95120
(spangles@almaden.ibm.com). Mr. Spangler is a Senior Technical Staff Member who has carried out research in knowledge-base and data mining for the past 15 years. He has developed software components for data visualization and text mining, which are available through eClassifier and Business Insights Workbench (BIW) service offerings. Mr. Spangler has a B.S. degree in mathematics from the Massachusetts Institute of Technology and an M.S. degree in computer science from the University of Texas.

Jeffrey T. Kreulen
IBM Research Division, Almaden Research Center,
650 Harry Road, San Jose, California 95120
(kreulen@almaden.ibm.com). Dr. Kreulen is a Senior Technical Staff Member and senior manager of the Services Oriented Technologies department. He has a B.S. degree in applied mathematics (computer science) from Carnegie-Mellon University, He also has an M.S. degree in electrical engineering and a Ph.D. degree in computer engineering, both from Pennsylvania State University. Dr. Kreulen has worked on multiprocessor systems design and verification, operating systems, systems management, Web-based service delivery, integrated text and data analysis, and the science of services.

James F. Newswanger
IBM Corporate Communications Group, 11 Madison Avenue, New York, New York 10010 (newswang@us.ibm.com). Dr. Newswanger is research manager of the IBM Corporate Intranet. He is responsible for the worldwide intranet measurement program at IBM, including site traffic analysis, employee satisfaction, and related qualitative research. He directs IBM Jam research, which has been recognized with an AMA/Nielsen EXPLOR award. Dr. Newswanger received M.S. and Ph.D. degrees, specializing in policy, research, evaluation and measurement (PREM), from the University of Pennsylvania. ■