The integration of business intelligence and knowledge management

by W. F. Cody
J. T. Kreulen
V. Krishna
W. S. Spangler

Enterprise executives understand that timely, accurate knowledge can mean improved business performance. Two technologies have been central in improving the quantitative and qualitative value of the knowledge available to decision makers: business intelligence and knowledge management. Business intelligence has applied the functionality, scalability, and reliability of modern database management systems to build ever-larger data warehouses, and to utilize data mining techniques to extract business advantage from the vast amount of available enterprise data. Knowledge management technologies, while less mature than business intelligence technologies, are now capable of combining today's content management systems and the Web with vastly improved searching and text mining capabilities to derive more value from the explosion of textual information. We believe that these systems will blend over time, borrowing techniques from each other and inspiring new approaches that can analyze data and text together, seamlessly. We call this blended technology BIKM. In this paper, we describe some of the current business problems that require analysis of both text and data, and some of the technical challenges posed by these problems. We describe a particular approach based on an OLAP (on-line analytical processing) model enhanced with text analysis, and describe two tools that we have developed to explore this approach—eClassifier performs text analysis, and Sapient integrates data and text through an OLAP-style interaction model. Finally, we discuss some new research that we are pursuing to enhance this approach.

A critical component for the success of the modern enterprise is its ability to take advantage of all available information. This challenge becomes more difficult with the constantly increasing volume of information, both internal and external to an enterprise. It is further exacerbated because many enterprises are becoming increasingly "knowledge-centric," and therefore a larger number of employees need access to a greater variety of information to be effective. The explosive growth of the World Wide Web clearly compounds this problem.

Enterprises have been investing in technology in an effort to manage the information glut and to glean knowledge that can be leveraged for a competitive edge. Two technologies in particular have shown good return on investment in some applications and are benefiting from a large concentration of research and development. The technologies are business intelligence (BI) and knowledge management (KM).

Business intelligence technology has coalesced in the last decade around the use of data warehousing and on-line analytical processing (OLAP). Data warehous-

ing is a systematic approach to collecting relevant business data into a single repository, where it is organized and validated so that it can be analyzed and presented in a form that is useful for business decision-making.[1] The various sources for the relevant business data are referred to as the operational data stores (ODS). The data are extracted, transformed, and loaded (ETL) from the ODS systems into a data mart. An important part of this process is data cleansing, in which variations on schemas and data values from disparate ODS systems are resolved. In the data mart, the data are modeled as an OLAP cube (multidimensional model), which supports flexible drill-down and roll-up analyses. Tools from various vendors (e.g., Hyperion, Brio, Cognos) provide the end user with a query and analysis front end to the data mart. Large data warehouses currently hold tens of terabytes of data, whereas smaller, problem-specific data marts are typically in the 10 to 100 gigabytes range.

Knowledge management definitions span organizational behavioral science, collaboration, content management, and other technologies. In this context, we are using it to address technologies used for the management and analysis of unstructured information, particularly text documents. It is conjectured that there is as much business knowledge to be gleaned from the mass of unstructured information available as there is from classical business data. We believe this to be true and assert that unstructured information will become commonly used to provide deeper insights and explanations into events discovered in the business data. The ability to provide insights into observed events (e.g., trends, anomalies) in the data will clearly have applications in business, market, competitive, customer, and partner intelligence as well as in many domains such as manufacturing, consumer goods, finance, and life sciences.

The variety of textual information sources is extremely large, including business documents, e-mail, news and press articles, technical journals, patents, conference proceedings, business contracts, government reports, regulatory filings, discussion groups, problem report databases, sales and support notes, and, of course, the Web. Knowledge and content management technologies are used to search, organize, and extract value from all of these information sources and are a focus of significant research and development.[2,3] These technologies include clustering, taxonomy building, classification, information extraction, and summarization. An increasing number of applications, such as expertise location,[4,5] knowledge portals, customer relationship management (CRM), and bioinformatics, require merging these unstructured information technologies with structured business data analysis.

It is our belief that over time techniques from both BI and KM will blend. Today's disparate systems will use techniques from each and will, in turn, inspire new techniques that will seamlessly span the analysis of both data and text. With this in mind, we describe our contributions in this direction. First, we briefly describe some business problems that motivate this integration and some of the technical challenges that they pose. Then we describe eClassifier, a comprehensive text analysis tool that provides a framework for integrating advanced text analytics. Next, we present an example that motivates our particular approach toward integrating data and text analysis and describe our architecture for a combined data and document warehouse and associated tooling. Finally, we discuss some current research directions in extracting information from documents that can increase the value of a data cube.

Motivation for BIKM

The desire to extend the capabilities of business intelligence applications to include textual information has existed for quite some time. The major inhibitors have included the separation of the data on different data management systems, typically across different organizations, and the immaturity of automated text analysis techniques for deriving business value from large amounts of text. The current focus on information integration in many enterprises is rapidly diminishing the first inhibitor, and advances in machine learning, information retrieval, and statistical natural language processing are eroding the second.

Examples of BIKM problems. To understand the importance of BIKM, it is useful to look at some real business problems and to determine how this technology can provide a return on the investment (ROI). The ROI can be achieved, in general, in one of two ways: (1) through cost reductions and identification of inefficiencies (improved productivity), and (2) through identification of revenue opportunities and growth. Here are some typical scenarios in which our customers believe their business analyses would benefit substantially from data and text integration:

1. *Understanding sales effectiveness*. A telemarketing revenue data cube can help identify products that

are most successfully sold over the phone, sales representatives who generate the most sales, and customers who are the most receptive to this sales approach. Unfortunately, the particular sales techniques used by these successful sales representatives in various situations are not captured by quantitative measures in the OLAP cube. However, these sales conversations are now frequently recorded and converted to text. The text of conversations associated with high-revenue sales representatives and high-yield customers can be analyzed by various language processing or pattern detection techniques to find patterns in the use of phrases or phrase sequences.

2. *Improving support and warranty analysis*. Frequently in business applications, short text descriptions, from, for example, customer complaints, are recorded in a database but are then encoded into short classification codes by a person. The code fields then become the basis for any business analysis of the set of customer complaints. Variations in the assignment of codes by different people can cause emerging trends or problem situations to be overlooked. The application of modern linguistic and machine-learning techniques (i.e., classification) to the text could provide a more consistent encoding, or at least a validation of the human encoding, as the basis for the business analysis.
3. *Relating CRM to profitability*. Data cubes for understanding revenues achieved over a set of customers frequently omit the costs associated with individual customers. In some industries these costs can substantially reduce the profit from a customer. The costs can include the number of calls the customer made into the business for problem resolution, complaint handling, or just "hand-holding." Extracting measures of these costs (e.g., time spent on the phone with the customer) and measures of the customer's loyalty for continued business (e.g., sentiment analysis of the customer interaction) from a customer relationship management (CRM) system and merging these measures into the revenue cube would provide a more complete picture of the profitability derived from a customer.[6]

Environmental issues. We have briefly presented some typical customer scenarios in which bringing text analysis together with classical data analysis can provide increased business value. Naturally, there are environments of varying complexity in which these scenarios occur, and consequently there are a variety of technologies and tools that may be needed in these different environments. In this section, we distinguish three general environments based on the degree of integration of the text and the data sources.

The simplest scenario occurs when the text information sits inside the same database as the business data and is unambiguously associated with the related business data. The text may simply be in character fields in the business data records or in separate tables linked with the data records through common join attributes. In this situation, text analysis techniques can be used to extract value from the text in the form of additional attributes, relationships, and facts that can then be directly related to the business data. As integrated database systems that bring text (e.g., XML [Extensible Markup Language]) together with data in a single database become more common, the ability to use text analysis to enrich the directly related data will also increase.

Currently, most textual information is in systems distinct from the ODS systems used to populate business intelligence data marts. In the simplest case the text system has meta-data that logically correspond to fields in the business data, for example, customer name or product name, which can be used to link the text with the data. However, the text system may use different forms for the meta-data values than those in the database, and this necessitates a data mapping transform to determine the correct association of text to data, for example, associating "DB2" with IBM DB2 Universal Database*, Enterprise-Extended Edition, or "J. Smith" with John W. Smith. These problems are common and difficult when integrating data from different source systems, but for this discussion we assume that enough data cleansing and transformation tools exist to at least somewhat automate this mapping.[7]

In the absence of adequate meta-data to relate the text to the data, classification technology can be used to categorize the text documents. The classes might correspond to the values in a data attribute—for example, the members in a dimension of an OLAP cube. The assignment of a document to a particular class for a data attribute (e.g., product name) could have a confidence measure associated with it and the document might be assigned to several classes. This classification process may require training, and it should make use of any relevant meta-data available in the database. Once the text has been appropriately related to the business data, it can be processed by the text analysis techniques to extract the desired bus-

iness information, such as additional attributes, relationships, or facts.[8]

A more complicated situation arises when, unlike in the previous examples, the sources of text to relate to a business data analysis are not known *a priori*. The relevant text sources can depend on the type of data analysis being performed, and the number of possibilities for such sources may be very large. In this case, a discovery process is needed to identify the appropriate text sources for the business analysis, and then an association technology is needed to relate the text to the data records. Finally, the appropriate text analysis can be used to extract the business value. As a brief example, consider a business analyst exploring a revenue cube and detecting a downward movement in revenues for a software product in some part of the United States.[9] The data cube shows the phenomenon but does not provide any explanation for it. Because the issue is the revenue decline of a software product in a certain region, there is a natural set of questions that might be asked to understand the revenue decline and a substantial number of text sources one might wish to review to find the answers. In general, the questions to be asked depend on the issue under investigation and the characteristics of the data, for example, its schema, its meta-data, its application context. In this example the text sources could include:

1. Enterprise-specific information, such as service call logs about the product and competitive intelligence reports about other companies' products
2. Purchased text information, from sources such as Hoovers and Dun & Bradstreet, on general software market conditions
3. Public documents in Web forums that contain discussions about products, such as ePinions.com

Current work on meta-data to represent the information content published in data sources and work on question-answering systems to match questions to information sources will help to automate the discovery process.[10] In all of these cases, the interactive analysis of data and text may ultimately require the use of a modern text-analysis tool to explore the text documents themselves. In the next section we describe such a tool.

eClassifier

Research and development investment in knowledge-management technologies has made significant progress. However, there still exists a need for an approach that integrates complementary and best-of-breed algorithms with guidance from domain expertise. eClassifier was designed to fill this need by incorporating multiple algorithms into an architecture that supports the integration of additional algorithms as they become proven. eClassifier is an application that can quickly analyze a large collection of documents and utilize multiple algorithms, visualizations, and metrics to create and to maintain a taxonomy. The taxonomies that eClassifier helps to create can be arbitrarily complex hierarchical categorizations. The algorithms and representation must be robust in order to apply across many diverse domains. In our research, we very quickly encountered environments where the documents to be analyzed were ungrammatical and contained misspellings, esoteric terms, and abbreviations. Help-desk problem tickets or discussion groups are examples of such environments.

eClassifier is a comprehensive text-analysis application that allows a knowledge worker to learn from large collections of unstructured documents. It was designed to employ a "mixed-initiative" approach that applies domain expertise, through interactions with state-of-the-art text analysis algorithms and visualization, to provide a global understanding of a document collection. Most of the complexities inherent in text mining are hidden by using default behaviors, which can be modified as a user gains experience. The tool can be used to automatically categorize a large collection of text documents and then provide to a knowledge worker a broad spectrum of controls to refine the building of an arbitrarily complex hierarchical taxonomy. eClassifier has implemented numerous analytical, graphical, and reporting algorithms to allow a deep understanding of the concepts contained within a document collection. The tool has been optimized to analyze over a million documents. Additionally, after a given taxonomy has been generated, a classification object can be published and used within another application, through the eClassifier run-time API (application programming interface), to dynamically retrieve information about the documents as well as to incrementally process new documents. Advanced visualization techniques allow the concept space to be explored from many different perspectives. Multiple taxonomies can be generated and explored to discover new relationships or important cross sections. Text sorting and extraction techniques provide valuable concept summarizations for each category. Many views are provided, including spreadsheets, bar graphs, plots, trees, and summary reports.

We have used eClassifier extensively in conjunction with Lotus Discovery Server* and IBM Global Services on both internal and customer information sources. Based on our application of eClassifier in various domains, with many users, we have reached the conclusion that it is very difficult to automatically produce a satisfactory taxonomy for a diverse set of users without allowing human intervention. The power of eClassifier is that it explicitly provides for the incorporation of human judgment at all appropriate phases of the taxonomy generation process. It provides the necessary tools for understanding the taxonomy, for efficiently editing it, and for validating that the taxonomy is learnable by a classifier.

Document representation. The applications for which eClassifier has typically been applied are characterized by documents about a single concept. Such application domains with documents that are relatively short include help-desk problem tickets and e-mail. In domains with longer, multitopic documents, some preprocessing is needed to break the documents down into conceptual chunks. Typically this is done using document structures such as chapters, sections, or paragraphs.

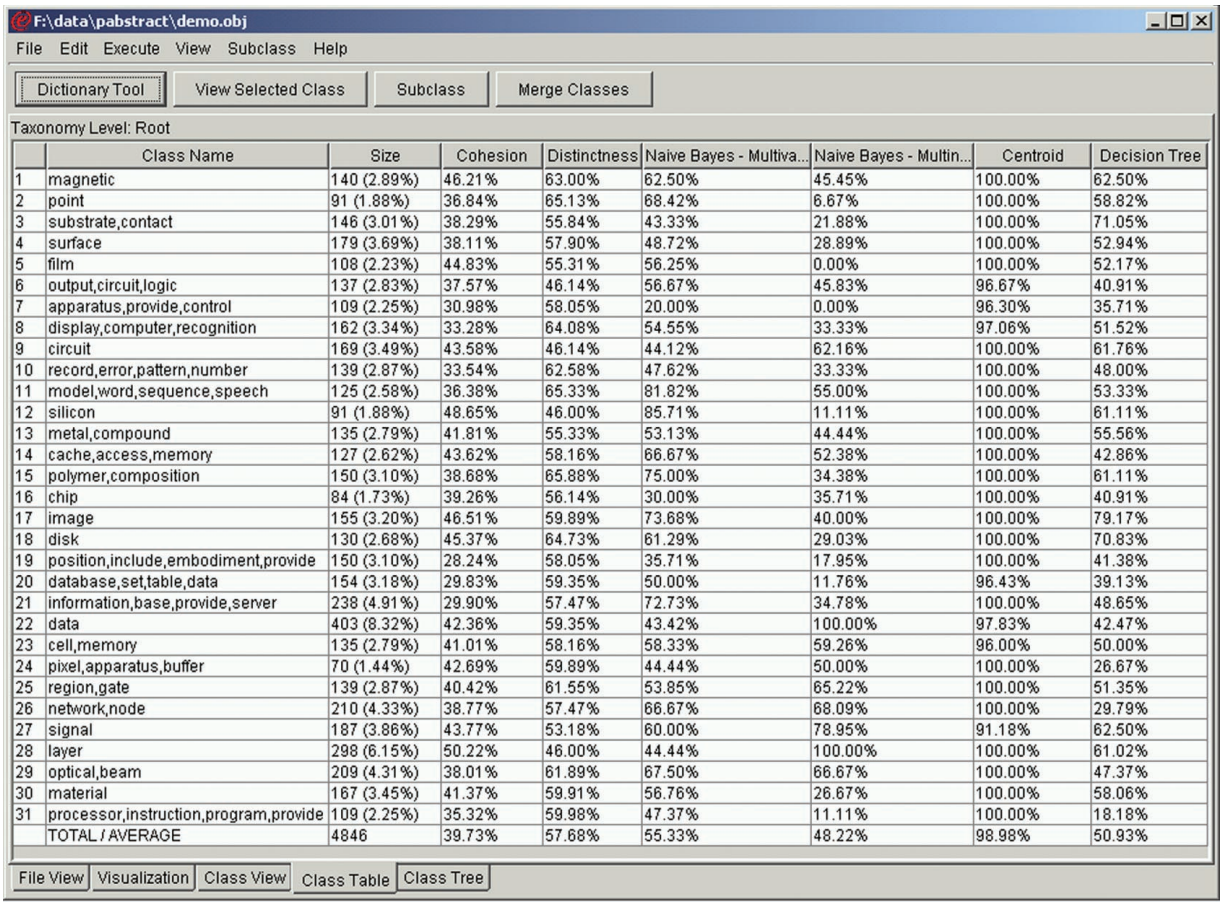
eClassifier represents each document with a vector of weighted frequencies from a feature space of terms and phrases.[11,12] The feature space is obtained by counting the occurrence of terms and phrases in each document and the vector is normalized to have unit Euclidean norm (the sum of the squares of the elements is one). To reduce the feature space representation for efficiency of computation and scalability, while maintaining maximum information, we utilize several techniques. We use stop-word lists to eliminate words bearing no content. We utilize synonym lists to collapse semantically similar words and stem variants to their base form. We use stock phrase lists to eliminate structural or no-content repetitive phrases. Stock phrases can also be automatically detected by the system through the use of statistical counting techniques. We use "include word" lists to identify semantically important terms that must remain in the feature space. Finally, we heuristically reduce the feature space by removing terms with the highest and lowest frequency of occurrences.

Once the feature space is determined, eClassifier uses a dictionary tool that provides a convenient method for dynamically inspecting and modifying the feature space. This tool provides a frequency measure and a relevance measure for each term and phrase. The frequency measure is the percentage of documents in which the term occurs, and the relevance measure is the maximum frequency with which a term occurs in any category, effectively measuring the term's influence on the taxonomy. Terms or phrases with high values for either of these measures should be considered carefully, because they heavily influence the document representation and therefore the resulting taxonomy. We have found this combination of techniques to be important and effective across a broad range of document sources.

Taxonomy generation. The first step in the analysis of the document collection is to create an initial categorization or taxonomy, which can be automated by applying clustering algorithms. In eClassifier we have implemented an optimized variant of the k-means algorithm[13,14] using a cosine similarity metric[15] to automatically partition the documents into k disjoint clusters. K-means can then be applied to each cluster to create a hierarchical taxonomy. In addition to k means we have implemented an EM (expectation maximization) clustering algorithm and EM with MHAC (modified hierarchical agglomerative clustering), which is a variant that generates hierarchical taxonomies.[16] In practice we have found automatic clustering algorithms to be very effective in creating initial taxonomies, which are used to get a sense of the concepts contained in the document collection. However, clustering does not always partition the documents in ways that are meaningful to a human. To partially address this, we have developed some additional methods for creating taxonomies, one of which is an interactive, query-based clustering that seeds categories based on a set of keywords, tests out the queries, and then refines the clusters based on the observed results. The query-based clusters can then be further subclassed using unsupervised clustering techniques. Finally, we have also found that it is sometimes useful to start with an initial classification based upon meta-data provided with the documents.

Taxonomy evaluation. Once we have an initial taxonomy of the documents, eClassifier provides the means to understand and to evaluate it. This allows us to address the unexpected results that do not meet human expectations. Figure 1 is an eClassifier screenshot showing summary information and statistics for a set of categories (note this could be at any depth in a hierarchical taxonomy). This view provides category label, size, cohesion, and distinctness measures. The vector-space model lends itself to computation of a normalized centroid for each cluster, which represents the average document in the cluster for the

Figure 1 eClassifier class table view



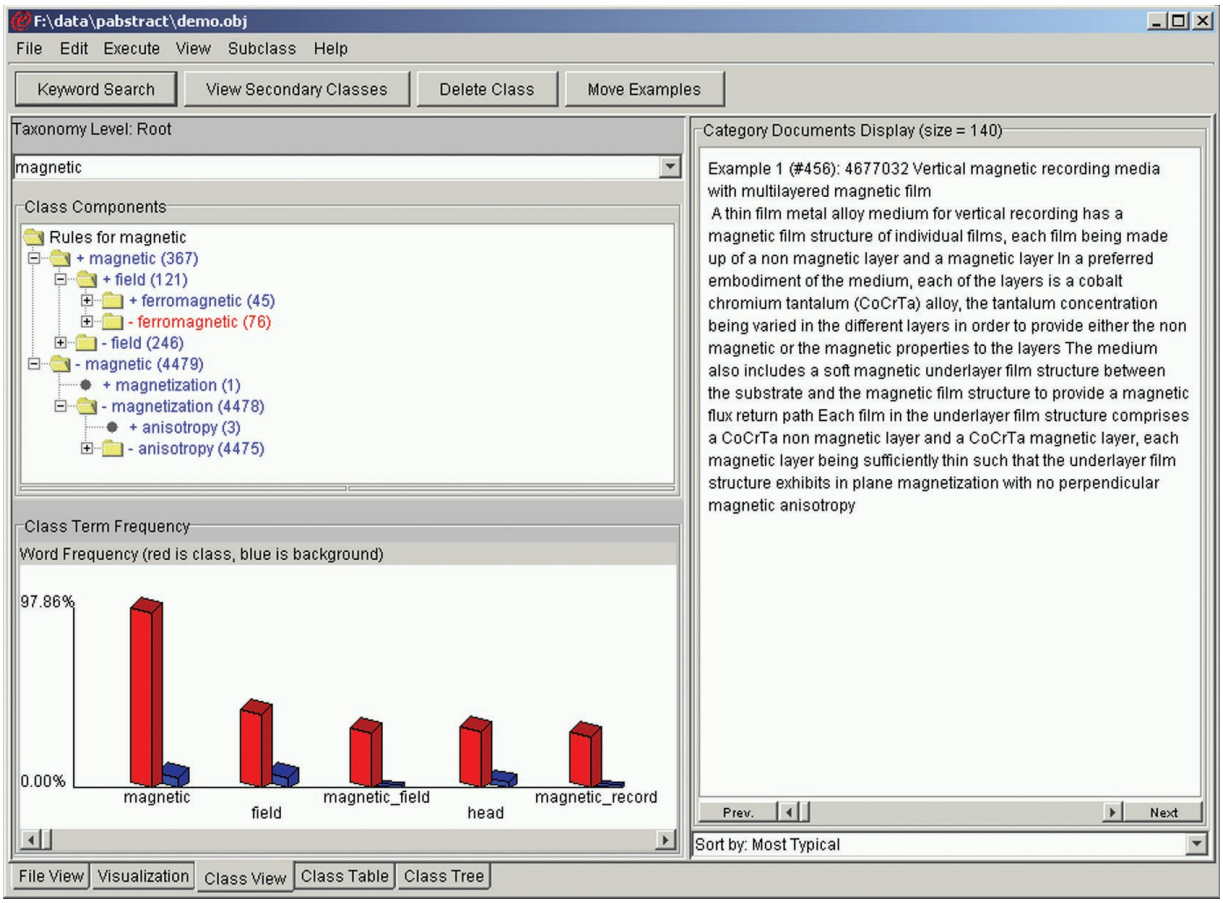
	Class Name	Size	Cohesion	Distinctness	Naive Bayes - Multiva...	Naive Bayes - Multin...	Centroid	Decision Tree
1	magnetic	140 (2.89%)	46.21%	63.00%	62.50%	45.45%	100.00%	62.50%
2	point	91 (1.88%)	36.84%	65.13%	68.42%	6.67%	100.00%	58.82%
3	substrate,contact	146 (3.01%)	38.29%	55.84%	43.33%	21.88%	100.00%	71.05%
4	surface	179 (3.69%)	38.11%	57.90%	48.72%	28.89%	100.00%	52.94%
5	film	108 (2.23%)	44.83%	55.31%	56.25%	0.00%	100.00%	52.17%
6	output,circuit,logic	137 (2.83%)	37.57%	46.14%	56.67%	45.83%	96.67%	40.91%
7	apparatus,provide,control	109 (2.25%)	30.98%	58.05%	20.00%	0.00%	96.30%	35.71%
8	display,computer,recognition	162 (3.34%)	33.28%	64.08%	54.55%	33.33%	97.06%	51.52%
9	circuit	169 (3.49%)	43.58%	46.14%	44.12%	62.16%	100.00%	61.76%
10	record,error,pattern,number	139 (2.87%)	33.54%	62.58%	47.62%	33.33%	100.00%	48.00%
11	model,word,sequence,speech	125 (2.58%)	36.38%	65.33%	81.82%	55.00%	100.00%	53.33%
12	silicon	91 (1.88%)	48.65%	46.00%	85.71%	11.11%	100.00%	61.11%
13	metal,compound	135 (2.79%)	41.81%	55.33%	53.13%	44.44%	100.00%	55.56%
14	cache,access,memory	127 (2.62%)	43.62%	58.16%	66.67%	52.38%	100.00%	42.86%
15	polymer,composition	150 (3.10%)	38.68%	65.88%	75.00%	34.38%	100.00%	61.11%
16	chip	84 (1.73%)	39.26%	56.14%	30.00%	35.71%	100.00%	40.91%
17	image	155 (3.20%)	46.51%	59.89%	73.68%	40.00%	100.00%	79.17%
18	disk	130 (2.68%)	45.37%	64.73%	61.29%	29.03%	100.00%	70.83%
19	position,include,embodiment,provide	150 (3.10%)	28.24%	58.05%	35.71%	17.95%	100.00%	41.38%
20	database,set,table,data	154 (3.18%)	29.83%	59.35%	50.00%	11.76%	96.43%	39.13%
21	information,base,provide,server	238 (4.91%)	29.90%	57.47%	72.73%	34.78%	100.00%	48.65%
22	data	403 (8.32%)	42.36%	59.35%	43.42%	100.00%	97.83%	42.47%
23	cell,memory	135 (2.79%)	41.01%	58.16%	58.33%	59.26%	96.00%	50.00%
24	pixel,apparatus,buffer	70 (1.44%)	42.69%	59.89%	44.44%	50.00%	100.00%	26.67%
25	region,gate	139 (2.87%)	40.42%	61.55%	53.85%	65.22%	100.00%	51.35%
26	network,node	210 (4.33%)	38.77%	57.47%	66.67%	68.09%	100.00%	29.79%
27	signal	187 (3.86%)	43.77%	53.18%	60.00%	78.95%	91.18%	62.50%
28	layer	298 (6.15%)	50.22%	46.00%	44.44%	100.00%	100.00%	61.02%
29	optical,beam	209 (4.31%)	38.01%	61.89%	67.50%	66.67%	100.00%	47.37%
30	material	167 (3.45%)	41.37%	59.91%	56.76%	26.67%	100.00%	58.06%
31	processor,instruction,program,provide	109 (2.25%)	35.32%	59.98%	47.37%	11.11%	100.00%	18.18%
	TOTAL / AVERAGE	4846	39.73%	57.68%	55.33%	48.22%	98.98%	50.93%

current feature space. The category centroid is central to the computation of the summary information in this view.

The category label is generated using a term-coverage algorithm that identifies dominant terms in the feature space. If a single term occurs in 90 percent or more of the documents in a category, the category is labeled with that term. If more than one term occurs with 90 percent frequency, then all of these terms (up to four) will be included in the name, with the "&" character between each term. If no single term covers 90 percent of the documents, then the most frequent term becomes the first entry in the name. The second entry is the one that occurs most frequently in all documents that do not contain the first term of the name. If the set of documents containing either of these two words is now 90 percent of the documents in the category, these two words combined become the name (separated by a comma). If not, this process is repeated. If none of the top four terms is contained in 10 percent or more of the documents, the category is called "Miscellaneous." We have experimented with various other algorithms for labeling categories, including finding the most frequently occurring phrases. Although these sometimes appear to be more meaningful, we have found them to be often misleading and to mischaracterize the category as a whole. Although this algorithm is effective for quickly summarizing a category, we also allow the user to assign a different label at any time.

In addition to a label, three metrics are computed for each category by default. The size column displays a count of the number of documents in the cat-

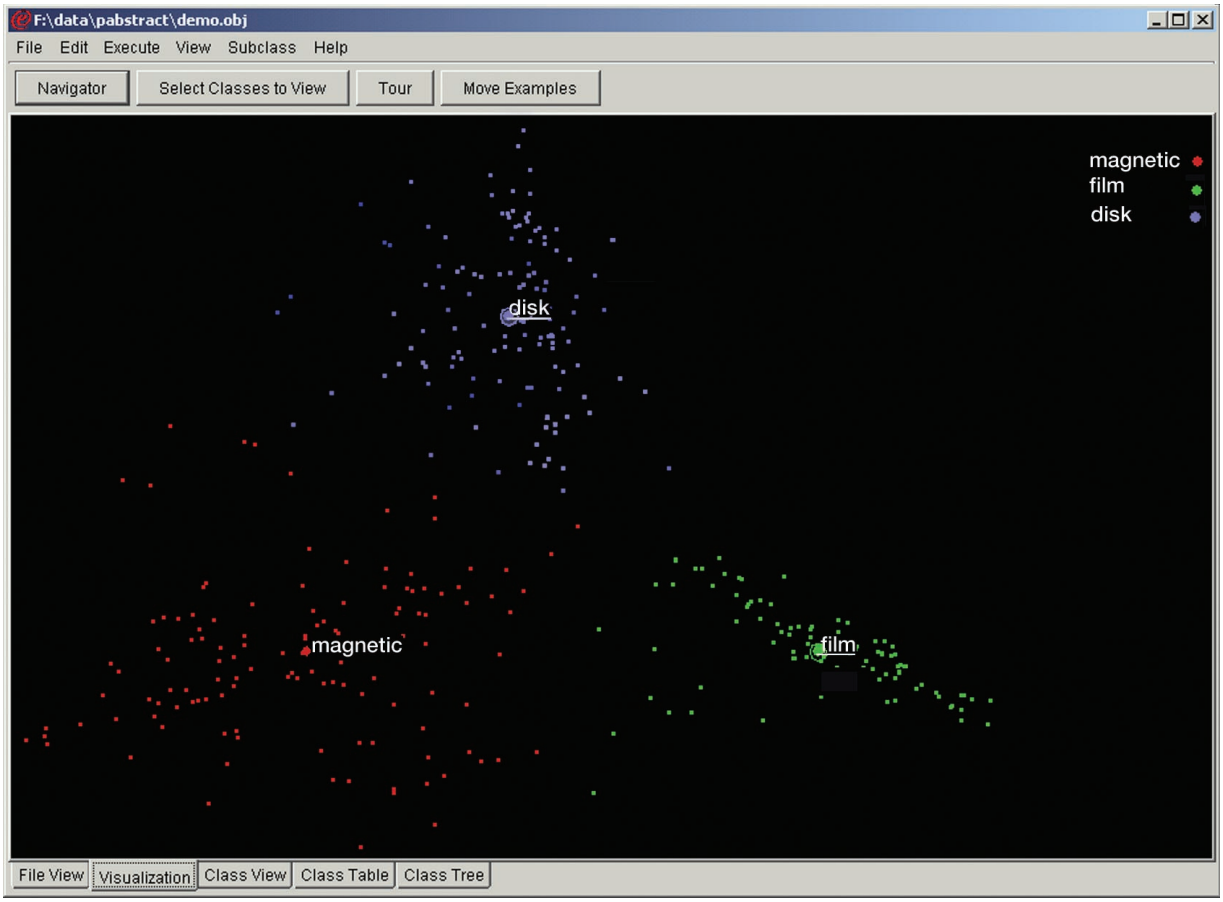
Figure 2 eClassifier class view



egory and its percentage of the total collection. Cohesion is a measure of the variance of the documents within a category. The cohesion is calculated based on the average cosine distance from the centroid. We have found that this provides a good measure of the similarity within a category, and categories with low cohesion are good candidates for splitting. Distinctness is a measure of variance of the documents between categories. The distinctness is calculated based on the cosine distance from a category's centroid to its nearest neighboring category's centroid. We have found this to provide a good measure of similarity between categories, so categories with low distinctness are similar to a neighboring category and would be candidates for potential merging.

Category evaluation. In addition to understanding a given taxonomy at a macro level, it is important to be able to precisely understand what core concept a category represents. To address this need, eClassifier provides a special view that shows statistics about term frequency, induced classification rules, and document examples as shown in Figure 2. eClassifier has a bar graph representation of the category centroid. For each term in the feature space, it shows the frequency of occurrence within the given class (red bar) and the frequency of occurrence within the total document collection (blue bar). The terms are sorted in decreasing order of red minus blue bar height in order to focus attention on the most relevant terms for the class. The class components panel visualizes the effect of certain terms (inclusion = +, exclusion = −) when used as a decision tree classifier. Nodes in the tree are selected based on minimizing the entropy of in-category vs out-of-category documents. If certain rules are par-

Figure 3 eClassifier visualization



ticularly meaningful, a user can click on the node and create a new class from the identified set of documents. Finally, this view provides example documents from each category. Several sorting techniques are available. Ordering by "most typical" is calculated based on proximity to the centroid. This is an effective technique—examining a few documents close to the centroid helps a user to understand the essence of the category's concept. Ordering by "least typical," by showing example documents farthest from the centroid, helps the user to evaluate uniformity within the category. Examples that the user identifies as not really belonging to the category can easily be moved to other categories or to a newly created category. With each modification, all relevant statistics are dynamically updated.

Taxonomy visualization. Visualization is an important technique to convey information. eClassifier uses visualization to help a user to explore the relationships between categories of documents within a taxonomy. We show an example of eClassifier's visualization in Figure 3. In the visualization each dot represents a document, which, when clicked on, will be displayed. The color of the dot denotes its membership within a corresponding category. A large dot represents a category centroid, which is the average feature vector for the documents in that category. For each rendering we select the centroids of three categories to form a plane. All the documents are then projected onto this plane.

This visualization is useful for exploring the relationship between various categories. We can quickly see which categories are close in proximity and we can find specific documents that may span these categories by selecting documents that lie on their respective borders.

The visualization gives multiple views of the data by allowing the user to select different planes on which to project. This can be done for all possible selections of three centroids to show many different views of the data, in procession. This process is known as touring.[17] The visualization also has a "navigator mode," which displays only closely related categories and allows the user to navigate by clicking on encircled centroids to show that category's most closely related categories.

Classification. Once a taxonomy is created for a document collection, it is often useful to assign additional documents to the taxonomy as they become available. To do this, eClassifier creates a batch classifier to process the additional documents. We have found that no single classification algorithm is superior under all circumstances, so we have implemented four algorithms and a methodology for evaluating which is best for a given document collection. For a given taxonomy, half of the documents are selected as the training set and half are left as the test set. A classification model is generated for each of the four implemented algorithms (nearest centroid, naive Bayes multivariate, naive Bayes multinomial, and decision tree) based on the training set. The best algorithm is then selected by determining classification accuracy performance on the test set. At each level of the taxonomy hierarchy a different classifier may be selected, based on which approach is most accurate at classifying the documents at that level. Additionally, as was the case during the clustering process, we allow complete control over the selection of the classification approach. Based on the (lack of) classification accuracy of the model selected, the user may choose to make adjustments to the taxonomy to improve the accuracy of the classifier. The classification accuracy for various classification algorithms can also be visualized in the class view (see Figure 1).

Analysis and reporting. In addition to the techniques described for taxonomy generation and maintenance, eClassifier provides several techniques for deeper analysis of the text, for example discovery of correlations of the text with corresponding data and for comparing document collections. The first technique we call "FAQ analysis" because it has commonly been applied to find frequently asked questions in help-desk data sets, although it can, in general, find frequently occurring topics in any document collection. Discovery of correlations is useful when analyzing a given taxonomy with respect to time (trend analysis) or against other associated meta-data. eClassifier will run a chi-squared test to find statistical anomalies for a given category in relation to other categorical attributes associated with documents. Continuous variables, such as time, are made discreet before analysis. Analyzing an attribute vs time in this way can lead to the discovery of spikes or other interesting trends. This technique can also be applied to any categorical data associated with the document. For example, assume we generated a technology-based taxonomy of patents using eClassifier. We could then analyze the patents to see which technologies are receiving the most patents over time. Once we know which technologies are "hot," we could then analyze the patents with their associated corporate assignees to see which corporations are active in the hot technologies.

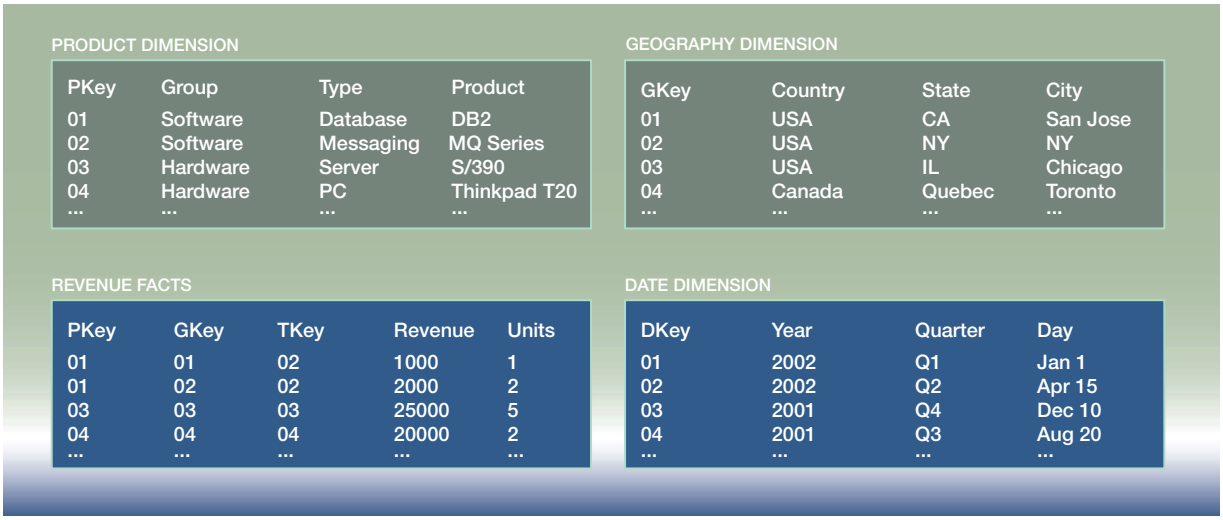
Another useful analysis is to use a generated taxonomy to compare document collections. For a given taxonomy and collection of documents, we can analyze a second collection of documents to discover which areas are poorly covered within the taxonomy. We have applied this technique to help-desk problem tickets and the associated self-help knowledge bases to identify knowledge gaps, for example, problems that are not well covered in the knowledge base. This can also be used to compare a collection of requirements documents against a collection of capability documents to discover knowledge-gap deficiencies.

An integration paradigm

In each of the environments discussed earlier, text is ultimately associated with business data records to enhance the understanding of the data. In some analysis-oriented environments, just bringing the associated text "near" the data with a flexible, interactive browsing and analysis tool such as eClassifier is sufficient to provide the user with some explanation for the business phenomenon. In the "discovery" environment this may be the natural and only realizable paradigm. Therefore, in addition to search capability, mechanisms to discover patterns, attributes, and schema in the documents, allowing them to be readily associated with the data, and tooling to provide an interactive analysis environment for both data and text will be helpful here. Though a valuable step, this approach has scalability problems if the number of sources or the number of associated documents is large.

In the more narrowly constrained first and second environments discussed earlier, we might strive to

Figure 4 Example star schema data model



achieve a tighter integration of the text information with the associated data. One way to do this is to use an OLAP multidimensional data model[1] as the integrating mechanism. The typical dimensional data model for an OLAP system uses a star schema as the model for a data cube. A basic star schema consists of a fact table at its center and a corresponding set of dimension tables, as shown in Figure 4. A fact table is a normalized table that consists of a set of measures or facts and a set of attributes represented by foreign keys into a set of dimension tables. The measures are typically numeric and additive (or at least partially additive). Because fact tables can have a very large number of rows, great effort is made to keep the columns as concise as possible. A dimension table is highly denormalized and contains the descriptive attributes of each fact table entry. These attributes can consist of multiple hierarchies as well as simple attributes. Because the dimension tables typically consist of less than 1 percent of the overall storage requirement, it is quite acceptable to repeat information to improve system query performance. The level at which the dimensions and measures encapsulate the data is referred to as the "fact grain." An example of a low-level grain is at the transactional level, where the dimensions are the product, geography, and date of the transaction, and the measures are the dollar revenue and units sold.

In the example in Figure 4 we have three dimension tables: product, geography, and date. The product dimension has an associated hierarchy: group → type → product. The geography dimension has an associated hierarchy: country → state → city. The date dimension has an associated hierarchy: year → quarter → day. These three dimensions represent the attributes that we can use to analyze our facts. In this example, we have a revenue fact table. Each row in the fact table represents the aggregate transactions at the lowest level in each of the dimensions. In this case, each fact is aggregated at product, city, and day. The measures that are aggregated are revenue and units sold. This model allows us to explore the effect of product, geography, and date, at all levels in each hierarchy, on revenue and units sold and other measures computed from these values such as total revenue, average revenue, total units sold, and average units sold. Typically an analyst would use an application to analyze these various measures by looking at trends over time, or by finding weaknesses or strengths in products or geographies.

Integrating text information into this analysis requires progress in several areas of text analytics in which we are currently working. The first is the use of text classification technology either to find attributes in the documents that can be used to link them to the data, or to find attributes in the documents that can be used as additional dimensions to deepen the understanding of the data. Second, we are researching information extraction technologies to process the text and to compute quantita-

Table 1 Schema for problem ticket documents

Product key	Geography key	Date key	Customer key	Problem type key	Days open	Severity of the complaint	Ticket identifier

tive values from the documents. The quantitative values can then be used as measures in a document fact table (Table 1). The combined data can not only be "sliced and diced" in the traditional OLAP paradigm of data analysis, but also the related documents can be explored in various ways that exploit their structure to make their content more useful. This interaction model and its underlying information model is an area for our current research.

Consider again the example in Figure 4. The facts have keys for the dimensions of product, geography, and date. Now we also have a database of problem tickets resulting from service calls. The problem tickets have meta-data recorded along with a transcript of the problem description. If we run a set of analyses over this collection of documents we can hope to accomplish several things. First, by using a classification process we can divide the problem tickets by problem type, thereby creating a new dimension, in addition to the existing meta-data dimensions, into which problem tickets can be categorized. Second, by running experimental text analyses over the text of the problem tickets we can attempt to quantify the severity of the problem in the ticket. Upon doing this, the problem ticket documents can be organized into their own fact table with the schema shown in Table 1.

The first four columns, which are foreign keys into dimension tables, are derived from the meta-data associated with the tickets in the problem ticket database. The fifth column is now a dimension associated with the problem that was created by automatically classifying the tickets. The sixth column is a measure associated with the problem that can be calculated from the meta-data. The seventh column is a measure of the severity of the problem as calculated by a text analysis of the transcription of the call. This may be a scoring of the frustration or anger felt by the caller. Finally, the last column ties this document fact back to the original document in the ticket database to facilitate movement from the OLAP environment of these facts into a document analysis environment.

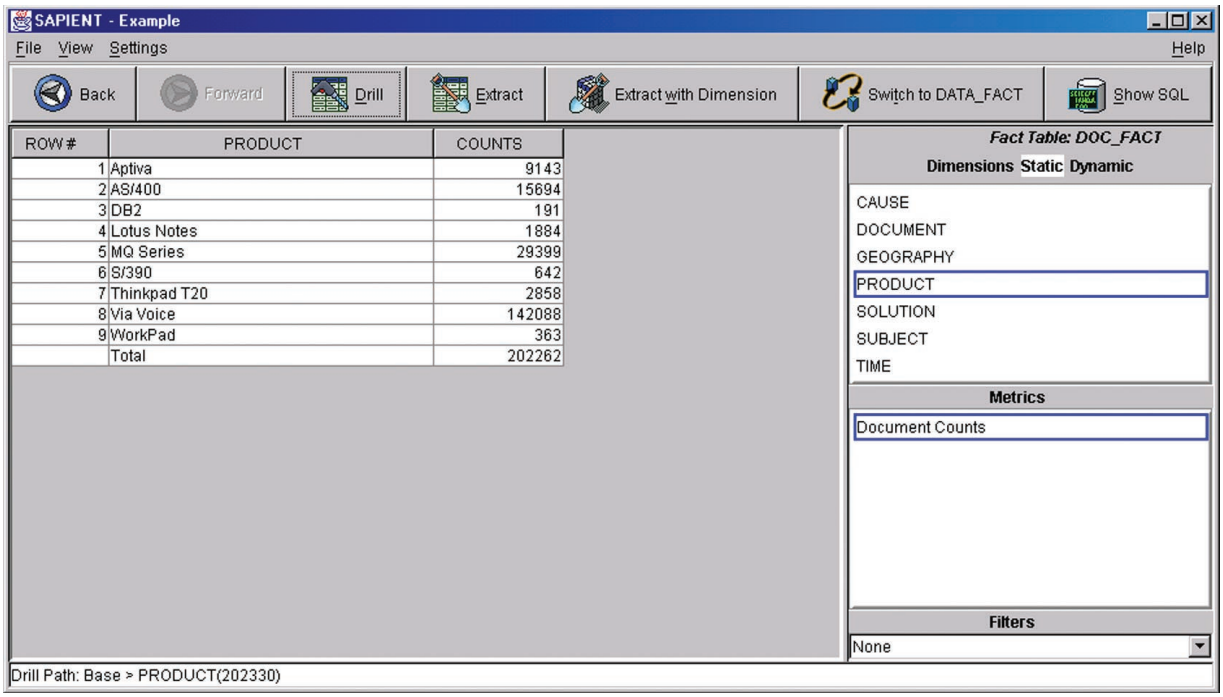
Given these fact table schemas, if we roll up the first fact table (Figure 4) along the product, geography, and date dimensions, while computing the average dollar sales and average units sold, and if we roll up the second table along the product, geography, date, and problem dimensions while computing the number of customer keys, the average days open, and the average complaint severity, then the join will give us a picture of the revenue as well as the problem costs for each product, per location, per time period associated with a problem type (e.g., installation, missing CDs, etc.). Then, with an integrated tooling environment we can perform this type of quantitative dimensional OLAP analysis and then seamlessly move into a document analysis to understand the complaints in more depth. A discussion of such an experimental tooling environment that has been built at the Almaden Research Center follows.

Integrated BIKM tools (Sapient & eClassifier). In the previous section we describe our text analysis system, eClassifier. In this section we describe the tooling we have built to apply the OLAP data model to text documents, creating a document warehouse. We then describe how we link the data model for the data and the documents through shared dimensions, and how this enhances our analytical capabilities. Finally, we describe how text analytics can be used to dynamically enhance this data model with what we call dynamic dimensions.

The tool we have developed allows us to explore data cubes with a star schema and consists of a report view and navigational controls. The report view provides a view of the results of data queries on a data cube. The reports can be summary tables (Figure 5), trend line graphs (Figure 6), or pie charts. An important part of the navigational controls are the dimensions and metrics selection boxes. The dimension selection box allows the user to select and drill down on each dimension. This includes drilling down a dimension hierarchy or cross drilling from one dimension to another. The metric selection box allows the user to select metrics that are computable for the given data cube. Additional navigation buttons allow forward and backward navigation to view previous reports. Other navigation controls are discussed later.

Document warehousing. We extend the techniques used on data in business intelligence to documents

Figure 5 Document counts for products



by using a dimensional model where the fact table granularity is a document, and the dimension tables hold the attributes of the document. Without additional processing this representation is a "factless" fact table, because there are, as yet, no associated measures. The process of populating the document warehouse has some complexities beyond typical ETL processing. In many cases the source of the documents is not an operational data store. Typically documents are automatically and incrementally put into the document warehouse based on either a subscription (push model) or a scheduled retrieval process (pull model). Additionally, we need a method to filter the documents because not all documents will be relevant to the purpose of the document cube.

Depending on the source, most documents have some associated meta-data that can naturally be used to populate some dimensions, such as author, date of publication, and document source. However, there are dimensions of potential interest that may not be included in the meta-data. If the dimension is known, classification techniques can be used to populate it. Using this model, all of the techniques previously described that are available to data cubes are now available to document cubes.

Shared dimensions. Thus far we have shown how star schemas can be used to organize and analyze both data and document cubes. Although each on their own can provide very useful information, providing a mechanism to link them will allow deeper analysis and thereby provide greater value. As an example, we revisit our product-geography-date revenue cube from Figure 4. If we have a collection of documents that are relevant to the given products, in the given geographies, over the given times, the information they contain and its relationship to the business data analysis can greatly improve decision making. Some documents that could provide insight in this example would be sales logs, customer support logs, news and press articles, marketing material, and discussion groups. All of these could provide unique insights into why a product is selling well or poorly in a given geography during a given time frame. The key to achieving these insights is to directly link the data to the documents through shared dimensions. An example data model of data and document cubes with shared dimensions is illustrated in Figure 7.

Dynamic dimensions. At this point we have data and document cubes that are linked through shared dimensions. All of the analytical techniques used on

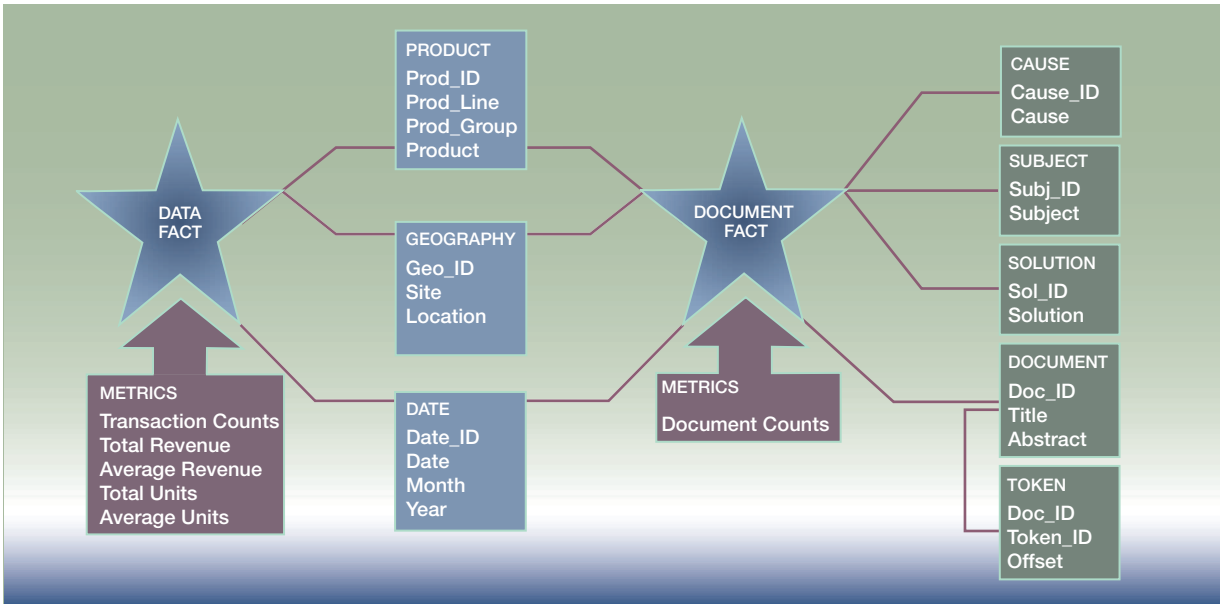
Figure 6 Time trend chart for products

data cubes can be used on the document cubes. Given the linkage created by shared dimensions, we can use the constraints used to identify a subset of data to then identify the corresponding set of documents and then make inferences from those documents about the data. For example, if the data show a drop in revenue for a product in certain geographies during a given time period, we can use these constraints on the document cube to identify the documents that might best explain the drop in revenue. We can then use standard OLAP techniques to investigate the relationship to any additional (non-shared) dimensions available for the documents. However, sometimes the existing dimensions and their taxonomies may be insufficient to fully explain the data. The documents can then be further analyzed using a deeper text analytical system such as eClassifier. We have provided this in our BIKM system by augmenting the document warehouse with an additional table (i.e., the token table) that has the document identifier, token identifier, and token offset for every token in every document (shown in Figure 7). The token table allows us to dynamically select (extract) and initiate eClassifier on an arbitrary subset of the documents from the document warehouse. Once we have invoked eClassifier on the documents we can perform all of the analytical capabilities outlined previously.

Furthermore, eClassifier can be used to create a new taxonomy over this selected set of documents. This new taxonomy is effectively a new (hierarchical) dimension that adds value to the existing data and document cubes. For example, problem tickets can be classified into problem types. This dimension provides a finer granularity for understanding the problems that are contributing to the costs associated with products in a given region and time period.

The new taxonomy can be made available to the document warehouse by creating a corresponding dimension table to represent the taxonomy and then populating an added column in the fact table, associating all known documents with the newly published dimension. This new dimension is now available to all of the analytical and reporting capabilities

Figure 7 Shared dimension data model



in the OLAP environment. Additional processing can be performed to classify all of the documents that were not in the extracted set of documents into the new dimension.

For example, we selected the "ThinkPad* T20" product (see Figure 5) and extracted into eClassifier the 2858 documents associated with this product. We used eClassifier to produce the new taxonomy shown in Figure 8. We then saved this taxonomy for the document warehouse by publishing it as the "new thinkpad taxonomy" dimension and updating the document fact table appropriately. This allows us to drill from within the data warehouse, and the results are shown in Figure 9.

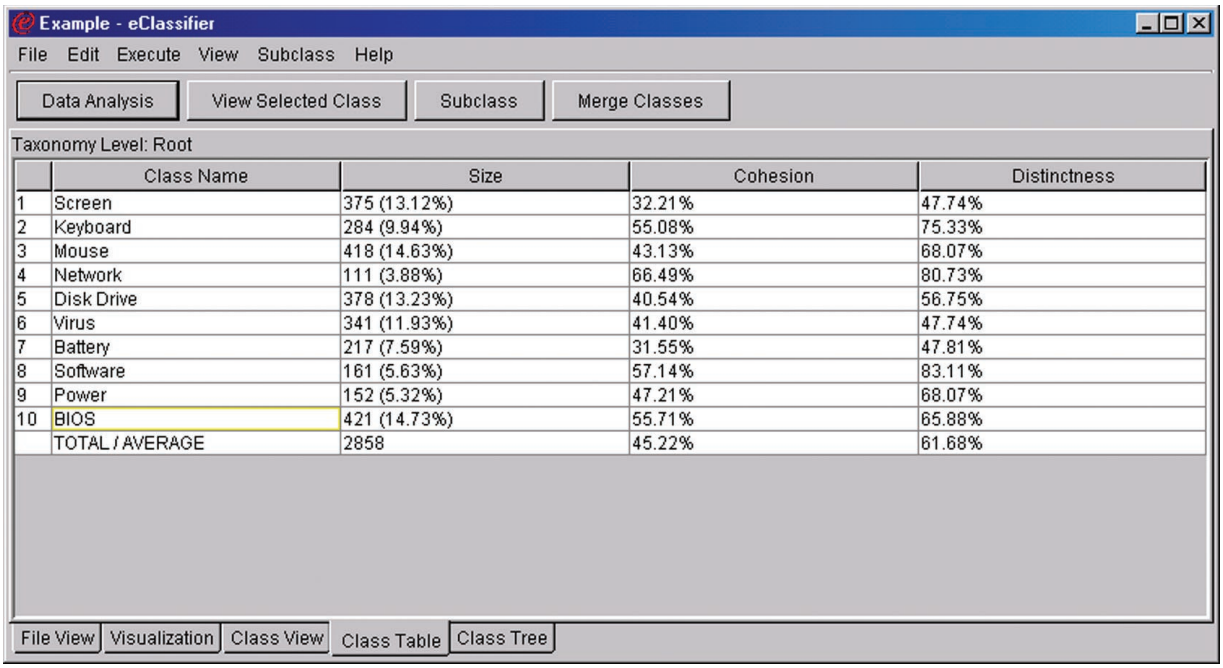
Summary and future research

The previous sections discuss our current integration model for data and text analysis and the tooling we have built to experiment with it. The missing, and somewhat open-ended, portion of this integration is the text analytics that will be used to create the quantitative metrics that populate the document cube and augment the data cube metrics. There is significant work going on in the IBM research community, especially within the unstructured information management area, to perform information extraction from documents. These efforts include: (1) extracting quantitative facts from documents (e.g., the financial terms of a contract); (2) deducing relationships between entities in a document (e.g., new product A competes with product B); and (3) measuring the level of subjective values such as severity or sentiment in documents (e.g., a customer letter reflects extreme displeasure with a company's service). Currently we are exploring techniques to accomplish these tasks based on statistical machine-learning approaches. We hope to report on these in a future paper.

Another area of future research that we believe is promising is the integration of ontologies into the taxonomy generation and dimension publishing portions of our BIKM architecture. Ontologies provide a level of semantics that we do not currently address, allowing improved taxonomies and reasoning about the data and text. Furthermore, emerging ontological technologies such as the semantic Web can provide a vehicle to integrate the text and data under study with a far larger body of text and data, thereby expanding the potential insights.

In this paper we show that text integrated with business data can provide valuable insights for improving the quality of business decisions. We describe a

Figure 8 eClassifier taxonomy for ThinkPad T20 documents



	Class Name	Size	Cohesion	Distinctness
1	Screen	375 (13.12%)	32.21%	47.74%
2	Keyboard	284 (9.94%)	55.08%	75.33%
3	Mouse	418 (14.63%)	43.13%	68.07%
4	Network	111 (3.88%)	66.49%	80.73%
5	Disk Drive	378 (13.23%)	40.54%	56.75%
6	Virus	341 (11.93%)	41.40%	47.74%
7	Battery	217 (7.59%)	31.55%	47.81%
8	Software	161 (5.63%)	57.14%	83.11%
9	Power	152 (5.32%)	47.21%	68.07%
10	BIOS	421 (14.73%)	55.71%	65.88%
	TOTAL / AVERAGE	2858	45.22%	61.68%

text analysis framework and how to integrate it into a business intelligence data warehouse by introducing a document warehouse and linking the two through shared dimensions. We believe that this provides a platform on which to build and research new algorithms to find the currently hidden business value in the vast amount of text related to business data. Technologies in the areas of information extraction and integrated text and data mining will build on this framework, allowing it to achieve its full business potential.

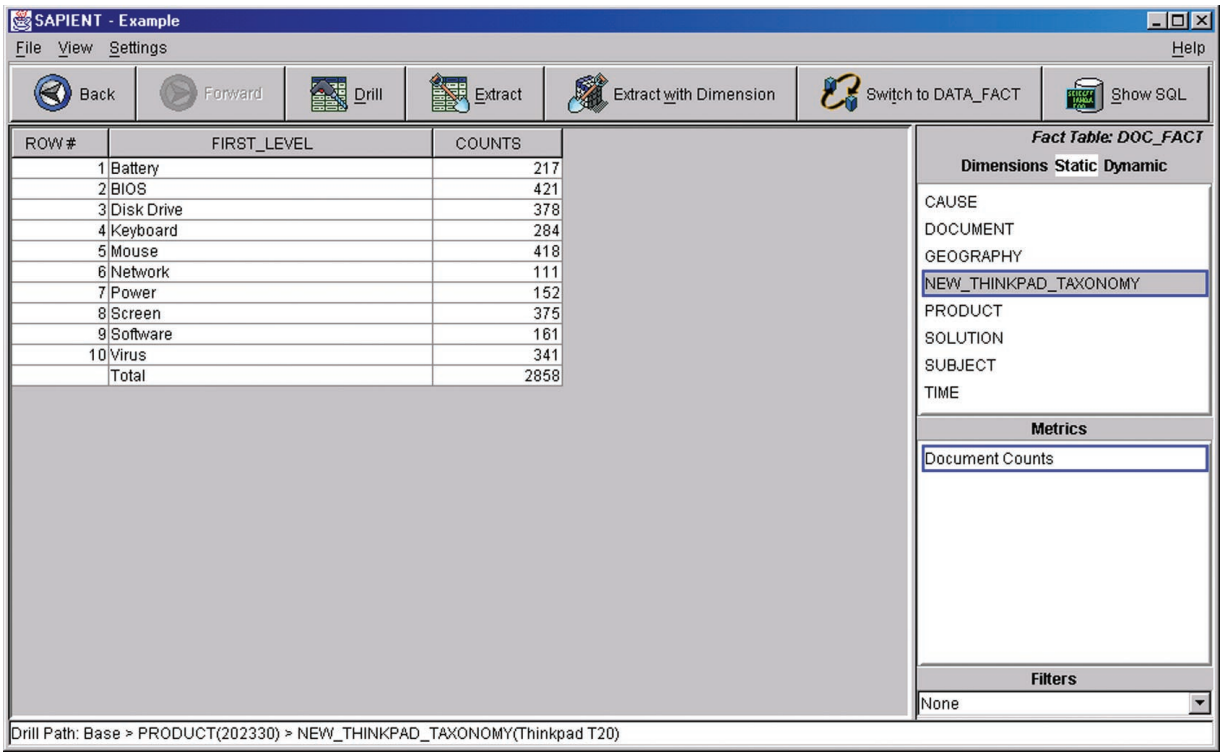
Acknowledgments

The authors gratefully acknowledge the contributions of Dharmendra Modha, Ray Strong, Justin Lessler, Thomas Brant, Iris Eiron, Hamid Pirahesh, Shivakumar Vaithyanathan, and Anant Jhingran for their contributions to eClassifier, Sapient, and the underlying ideas of BIKM.

Cited references

1. R. Kimball, *The Data Warehouse Toolkit*, John Wiley & Sons, Inc., New York (1996).
2. D. Sullivan, *Document Warehousing and Text Mining*, John Wiley & Sons, Inc., New York (2001).
3. T. Nasukawa and T. Nagano, "Text Analysis and Knowledge Mining System," *IBM Systems Journal* **40**, No. 4, 967–984 (2001).
4. W. Pohs, *Practical Knowledge Management*, IBM Press, Double Oak, TX (2001).
5. W. Pohs, G. Pinder, C. Dougherty, and M. White, "The Lotus Knowledge Discovery System: Tools and Experiences," *IBM Systems Journal* **40**, No. 4, 956–966 (2001).
6. See http://www-4.ibm.com/software/data/bi/banking/ezmart.htm.
7. M. Hernandez, R. J. Miller, and L. Haas, "Clio: A Semi-Automatic Tool for Schema Mapping," *Proceedings*, *Special Interest Group on Management of Data*, Santa Barbara, CA (May 21–24, 2001).
8. See http://www.itl.nist.gov/iad/894.02/related_projects/muc/index.html.
9. S. Sarawagi, R. Agrawal, and N. Megiddo, "Discovery-Driven Exploration of OLAP Data Cubes," *Proceedings*, *6th International Conference on Extending Database Technology*, Valencia, Spain (March 23–27, 1998), pp. 168–182.
10. C. Kwok, O. Etzioni, and D. S. Weld, "Scaling Question Answering to the Web," *Proceedings*, *10th International World Wide Web Conference*, Hong Kong (May 1–5, 2001), available at http://www10.org/cdrom/papers/120/.
11. G. Salton and M. J. McGill, *Introduction to Modern Retrieval*, McGraw-Hill Publishing, New York (1983).
12. G. Salton and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval," *Information Processing and Management* **4**, No. 5, 512–523 (1988).
13. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, Inc., New York (1973).

Figure 9 Dynamic dimension results



14. J. A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, Inc., New York (1975).
15. E. Rasmussen, "Clustering Algorithms," W. B. Frakes and R. Baeza-Yates, Editors, *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, Englewood Cliffs, New Jersey (1992), pp. 419–442.
16. S. Vaithyanathan and B. Dom, "Model-Based Hierarchical Clustering," available at http://www.almaden.ibm.com/cs/people/dom/papers/uai2k.ps.
17. I. Dhillon, D. Modha, and S. Spangler, "Visualizing Class Structures of Multi-Dimensional Data," *Proceedings*, *30th Conference on Interface, Computer Science and Statistics*, May 1998.

Accepted for publication July 12, 2002.

William F. Cody *IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, California 95120 (electronic mail: wcody@almaden.ibm.com).* Dr. Cody is a senior manager of the Knowledge Middleware and Technology group at IBM's Almaden Research Center. He received his Ph.D. degree in mathematics in 1979 and has held various product development, research, and management positions with IBM since joining the company in 1974. He has published papers on database applications, database technology, software engineering, and group theory.

Jeffrey T. Kreulen *IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, California 95120 (electronic mail: kreulen@almaden.ibm.com).* Dr. Kreulen is a manager at the IBM Almaden Research Center. He holds a B.S. degree in applied mathematics (computer science) from Carnegie Mellon University and an M.S. degree in electrical engineering and a Ph.D. degree in computer engineering, both from Pennsylvania State University. Since joining IBM in 1992, he has worked on multiprocessor systems design and verification, operating systems, systems management, Web-based service delivery, and integrated text and data analysis.

Vikas Krishna *IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, California 95120 (electronic mail: vikas@almaden.ibm.com).* Mr. Krishna is a software engineer at the IBM Almaden Research Center. He holds a B.Tech. degree in naval architecture from IIT Madras, an M.E. degree in computational fluid dynamics from Memorial University, Newfoundland, Canada, and a M.S. degree in computer engineering from Syracuse University, New York. Since joining IBM in 1997, he has developed systems for Web-based service delivery, business-to-business information exchange, and the integrated analysis of text and data.

W. Scott Spangler *IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, California 95120 (electronic mail:*

spangles@almaden.ibm.com). Mr. Spangler has been doing knowledge base and data mining research for the past 15 years—lately at IBM and previously at the General Motors Technical Center, where he won the prestigious "Boss" Kettering award (1992) for technical achievement. Since coming to IBM in 1996, he has developed software components, available through the Lotus Discovery Server product and IBM alphaWorks®, for data visualization and text mining. He holds 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.