Generating and Browsing Multiple Taxonomies Over a Document Collection

SCOTT SPANGLER, JEFFREY T. KREULEN, AND JUSTIN LESSLER

SCOTT SPANGLER has been doing knowledge base and data mining research for the past 15 years—recently at the IBM Almaden Research Center and previously at the General Motors Technical Center. Since coming to IBM in 1996, he has developed software components for data visualization and text mining, which are available through the Lotus Discovery Server product and IBM Alphaworks. Mr. Spangler has published papers at ACM-SIGKDD, *Machine Learning,* IAAI, ACM Hypertext, and HICSS. He holds five patents and has several more patents pending. Scott Spangler holds a B.S. in Math from MIT and an M.A. in Computer Science from the University of Texas.

JEFFREY T. KREULEN is a manager at the IBM Almaden Research Center. He holds a B.S. in applied mathematics (computer science) from Carnegie Mellon University, and an M.S. in electrical engineering and a Ph.D. in computer engineering from the 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.

JUSTIN LESSLER started his career with IBM after graduating the University of North Carolina at Chapel Hill in 1996 and has been with the IBM Almaden Research Center since 1999.

ABSTRACT: We present a novel system and methodology for generating and then browsing multiple taxonomies over a document collection. Taxonomies are generated using a broad set of capabilities, including meta data, key word queries, and automated clustering techniques that serve as a seed taxonomy. The taxonomy editor, eClassifier, provides powerful tools to visualize and edit each taxonomy to make it reflective of the desired theme. Cluster validation tools allow the editor to verify that documents received in the future can be automatically classified into each taxonomy with sufficiently high accuracy.

In general, those seeking knowledge from a document collection may have only a vague notion of exactly what they are attempting to understand, and would like to explore related topics and concepts rather than simply being given a set of documents. For this purpose, we have developed MindMap, an interface utilizing multiple taxonomies and the ability to interact with a document collection.

KEY WORDS AND PHRASES: data mining, document classification, document clustering techniques, knowledge management, navigation, taxonomy, text mining, visualization.

Journal of Management Information Systems / Spring 2003, Vol. 19, No. 4, pp. 191–212.

0742–1222 / 2003 $9.50 + 0.00.

BUSINESSES HAVE BEEN ABLE TO systematically increase the leverage gained from enterprise data through technologies such as relational database management systems and techniques such as data warehousing. Moreover, it is conjectured that the amount of knowledge encoded in electronic text far surpasses that available in data alone. However, the ability to take advantage of this wealth of knowledge is just beginning to meet the challenge. Businesses that can take advantage of this potential will surely gain a competitive advantage through better decision-making and increased efficiencies [4]. One important step in achieving this potential has been to structure the inherently unstructured information in meaningful ways. A well-established first step in gaining understanding is to segment examples into meaningful categories [2, 4]. This leads to the idea of taxonomies—natural hierarchical organizations of the information in alignment with the business goals, organization, and processes. Whereas there will be some commonality in some industries, these natural organizations will have significant diversity across domains and organizations. Even with a single organization, no one taxonomy may capture all the important relationships between documents. Multiple taxonomies are needed to fully capture document meta-knowledge.

Research to address this need for taxonomy development has concentrated largely around automated grouping techniques such as text clustering. Text clustering is a set of techniques for automatically grouping together documents that contain many common words. A dictionary, or feature space, is generated by parsing the documents into tokens, pruning out the tokens that occur in a known stop words list, and combining the tokens that occur in a synonym list. Whereas we believe that text clustering is an invaluable tool, indeed it is part of our solution, we assert that it is insufficient to meet the full challenge of taxonomy generation by itself. Our experience using variations of *k*-means [11, 20] (i.e., a distance-based clustering technique where the number of clusters, *k*, is specified up front and the algorithm chooses a random set of *k* seeds; instances are assigned to the nearest centroid; this causes the centroid to move when recomputed as the average of points in that cluster and the process then repeats until all centroids are stable) assigned and Expectation Maximization (EM) clustering algorithms [23] (i.e., a probability-based approach that assumes as Gaussian or normal distribution for each cluster and then iteratively estimates the mean, standard deviation, and cluster probability for each cluster) have shown that they generate useful seed taxonomies, but rarely generate a satisfactory final taxonomy for a given business problem. For example, if you were to cluster a set of patents with the intent to create a technology-based taxonomy you would typically find some of the clusters to be technologies and some to be based on some other aspect or relationship. One might postulate that the clustering algorithm is in fact not the issue, but that this is a feature selection problem. With this in mind, an alternative approach would be to leverage controlled vocabularies. However, we find this approach to be very labor-intensive and would still yield results that would need further refinement.

Our approach to solve this problem focuses on the visualization, editing, and validation of clustering results. Clearly, it is not practical to read each document and categorize it, however, expert inspection guided by appropriate feedback is a powerful

combination. Clusters can be seeded with examples selected via key words in order to generate multiple taxonomies, each with a different theme. We have implemented these capabilities in eClassifier and describe how multiple taxonomies may be generated in the second section, edited in the third section, and validated in the fourth section.

Once we have developed multiple taxonomies that are relevant to a document collection, it is paramount that the tools available to understand and explore the information are effective. Currently, the most popular methods are a combination of Boolean key word searching and the use of a single taxonomy, best represented by the Internet search engines Yahoo! and Google. These techniques still fall short in the face of the many ambiguities, complexities, and domain-specific relationships that are contained in documents.

To address these deficiencies, we present a novel system and methodology for browsing and exploring topics and concepts within a document collection. Our system leverages multiple taxonomies, related terms, visualization, and user interaction to navigate and explore a document collection and the concepts that it contains. We call our system MindMap because we have modeled our interface on techniques used for brainstorming.

The techniques developed have been found to be particularly useful when exploring a complex topic that is not yet fully understood by the user. The techniques bring to light related concepts and terms that help round out the understanding, whereas still allowing the user to get to specific documents and delve into the detail needed for in-depth understanding.

Multiple Taxonomy Generation

BEFORE A DOCUMENT COLLECTION CAN BE EXPLORED using the MindMap interface, it must be categorized into multiple taxonomies. We use eClassifier to generate multiple taxonomies, where each taxonomy is designed around a specific theme. The purpose of each taxonomy is to group related documents together in order to present a user with sets of documents that share common characteristics. Taxonomies generated using characteristics that bring to light interesting relationships are always more enlightening and tend to be domain- and application-specific. Some interesting example characteristics that we have come across are industry, geography, technology, document source, process stage, and document creation time, but there are many more. The need for multiple taxonomies arises because in many cases there exists no single taxonomy that captures all interesting relationships between documents, and users will approach an investigation with different prior knowledge and goals.

For example, assume we have a set of documents containing a brief description of the Fortune 500 companies with one document per company. Each document may describe what the company produces, where the headquarters are located, what technologies the company has leadership in, and what business partnerships each company has. In other words, several taxonomies could be created over a single set of documents, each with a different theme. An example is shown in Figure 1.

Geography
North America
Europe
Africa
Asia
...

Technology
Chemical Engineering
Computer Networking
Alternative Fuels
Software
...

Industry
Energy
Transportation
Computers
Services
...

Figure 1. Example of Thematic Taxonomies

Each of these taxonomies provides a unique way of defining what it means to be "similar." In the "geography" context, to be similar means to be located in the same geographic region, whereas in the industry context, it means to be competitors in the market. Each of these taxonomies is valuable in some information retrieval context.

Many approaches may be used to generate multiple taxonomies over a document collection. Text clustering over a feature space of term occurrence within documents is one way to generate a generic content-based taxonomy. After eliminating common stop words and (high- and low-frequency) non-content-bearing words, we represented the text data set as a vector space model. That is, we represented each document as a vector of certain weighted frequencies of the remaining words [22] using the txn weighting scheme [21]. This scheme emphasizes words with high frequency in a document, and normalizes each document vector to have unit Euclidean norm. We have found the k-means algorithm [7] to be an effective tool for generating a high-level taxonomy over a collection of short documents. Whereas we have found *k*-means to be useful, different clustering methods can be used to create different taxonomies (see [8, 11, 20]).

Additional taxonomies may be generated by starting from a key word description of each category in the taxonomy. These key words are then included a priori as terms in the vector space dictionary for that taxonomy (thus dictionaries can and often do vary with each taxonomy). An initial classification of the documents is then generated by selecting for each document the category with which it shares the most key words. Documents containing no key words can be placed in a "miscellaneous" category. After this initial classification by key words is completed, nearest centroid methods may be employed to categorize some or all of the examples in the miscellaneous category into one of the user-defined categories.

Whereas we have found text clustering and key word search to be useful, there are many alternative methods for creating taxonomies, from document usage patterns [3] to manual analysis of business documents [14]. The methods described below are equally useful in such cases.

Before a user can begin editing taxonomy, they must first fully understand the existing categories and their relationships. In this section, we describe the strategy employed by our taxonomy-editing tool, eClassifier, to communicate the features of a document taxonomy to the user.

Summaries

Since we cannot expect the user to spend the time to read through the individual documents in a category, summarization is an important tool in helping the user understand what a category contains. Summarization techniques based on extracting text from the individual document [12, 13] were found to be insufficient in practice for the purpose of summarizing an entire document category, especially when the theme of that category is somewhat diverse. Instead, we employ two different visual 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 in the category. Each entry consists of two bars, a red bar to indicate what percentage of the documents in the category contain the feature, and a blue bar that indicates how frequently the feature occurs in the background population of documents from which the category was drawn. The bars of the chart are sorted in decreasing order of the difference between blue and red. Thus, the most important features of the category in question are shown at the beginning of the chart. This chart quickly summarizes for the user all the important features of a category, with their relative importance indicated by the size of the bars.

The second technique is a dynamic decision tree representation that describes what feature combinations define the category. This tree is generated in the same manner as a binary ID3 tree [18], selecting at each decision point the attribute that is most helpful in splitting the whole population of documents so that the two new categories of documents created are most nearly pure category and pure non-category. Each feature choice is made on-the-fly as the user expands each node until no additional features will improve the purity with respect to the category. The result is 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 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 is a node whose membership is more than (or equal to) 50 percent in category, blue is a node whose membership is less than 50 percent in category. This display gives the user an in-depth definition of the category in terms of salient features and lets them readily select various category components for further study (see Figure 2).

Visualization

In order to understand specifically how two or more categories at the same level of the taxonomy relate to each other, a visualization strategy is employed. The idea is to visually display the vector space of a bag-of-words document model [21, 22] to the user so that the documents will appear as points in space. The result is that documents containing similar words occur near to 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 a point on an X,Y scatter plot. The difficulty is that the dimensions of the document vector space will be of a much higher dimension. In fact, the

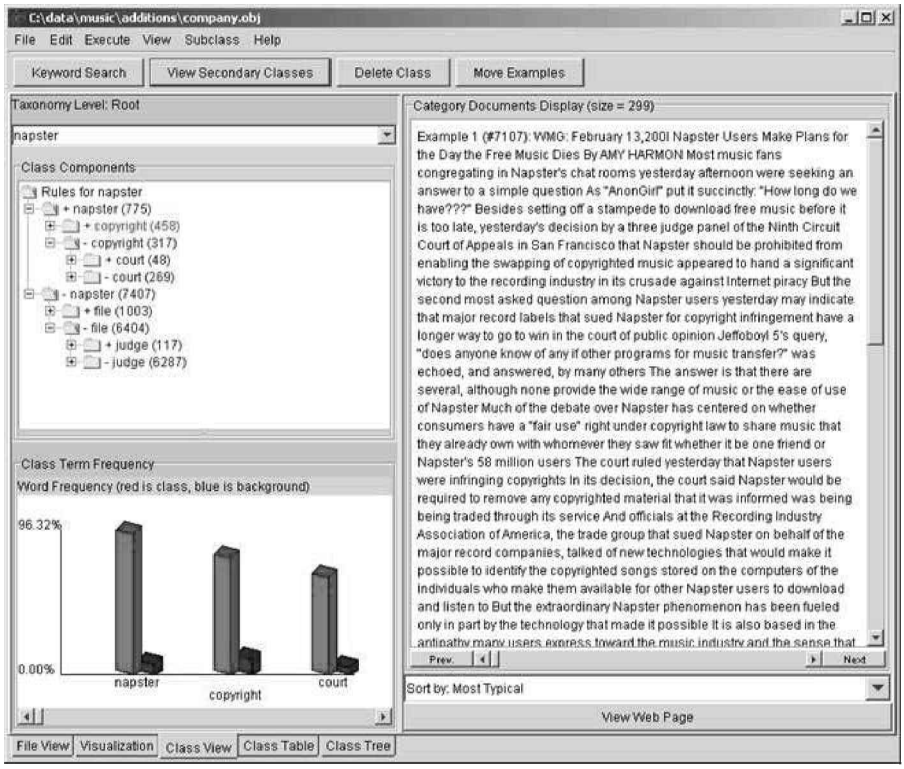


Figure 2. Category Summarization

dimensionality will be 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 [6], which relies on three category centroids to define the plane of most interest and to project the documents as points on to 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 colored according to category membership. The centroid of the category is also displayed. The resultant plot is a valuable way to discover relationships between neighboring concepts in a taxonomy (see Figure 3).

Sorting of Examples

When studying the examples in a category to understand the category's essence, it is important that the user not have to select the examples at random. To do so can sometimes lead to a skewed understanding of the content of a category, especially if the sample is small compared to the size of the category (which is often the case in prac-

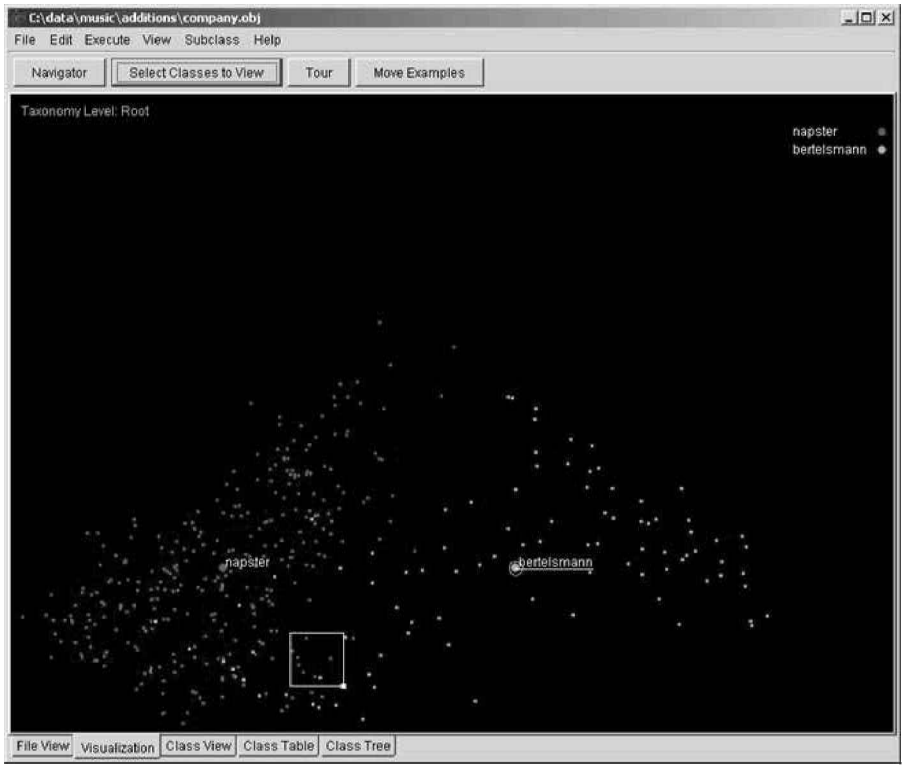


Figure 3. Floating Box for Moving Documents

tice). To help alleviate this problem, eClassifier allows sorting of examples based on the "most typical" first or "least typical" first criteria. This translates in vector space terms to sorting in order of distance from the category centroid (i.e., most typical is closest to the centroid, least typical is furthest from the centroid). The advantage of sorting in this way is twofold: reading documents in most typical order can help the user to quickly understand what the category is generally about, without having to read a large sample of the documents in the category, whereas reading the least typical documents can help the user to understand the total scope of the category and if there is conceptual purity.

Editing the Taxonomy

ONCE THE CONTENT MANAGER UNDERSTANDS the meaning of the categories in the taxonomy and their relationship to each other, the next step is to provide tools for rapidly changing the taxonomy to reflect the needs of the application. Keep in mind that our goal here is not to produce a "perfect" taxonomy for every possible purpose. Such a taxonomy may not even exist, or at least may require too much effort to obtain. Instead we want to focus the users efforts on creating a "natural" taxonomy

that is practical for a given application. For such applications, there is no right or wrong change to make. It is important only that the change accurately reflect the expert user's point of view about the desired structure. In this situation, the user is always right. The tool's job is to allow the user to make whatever changes may be deemed desirable. 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 taxonomy editing tool, eClassifier, provides capabilities at every level of a taxonomy to allow the user to make the desired modifications with a simple point and click.

The distance metric employed to compare documents to each other and to the category centroids is the cosine similarity metric [21]. However, as will be seen, during the category editing process we are not rigid in requiring each document to belong to the category of its nearest centroid, nor do we strictly require every document to belong to only one category.

Category Level

Category level changes involve modifying the taxonomy at a macro level, without direct reference to individual documents within each category.

Merging

Merging two categories 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 supplies the new category with an appropriate name.

Deleting

Deleting a category (or categories) means removing the category from the taxonomy. The user needs to recognized this may have unintended consequences, since all the examples that formerly belonged to the deleted category must now be placed in a different category—the one having the nearest centroid. To make this decision more explicit, we introduce the graphic called "View Similar Categories" chart.

The chart in Figure 4 displays what percentage of a categories documents would go to which other categories if the selected category were to be deleted. Each slice of the displayed pie chart can be selected to view the individual documents represented by the slice. Making such information explicit allows the user to make an informed decision when deleting a category, avoiding unintended consequences.

Document Level

Whereas some changes to a taxonomy may be made at the category level, others require a finer degree of control. These are called document level changes, and con-

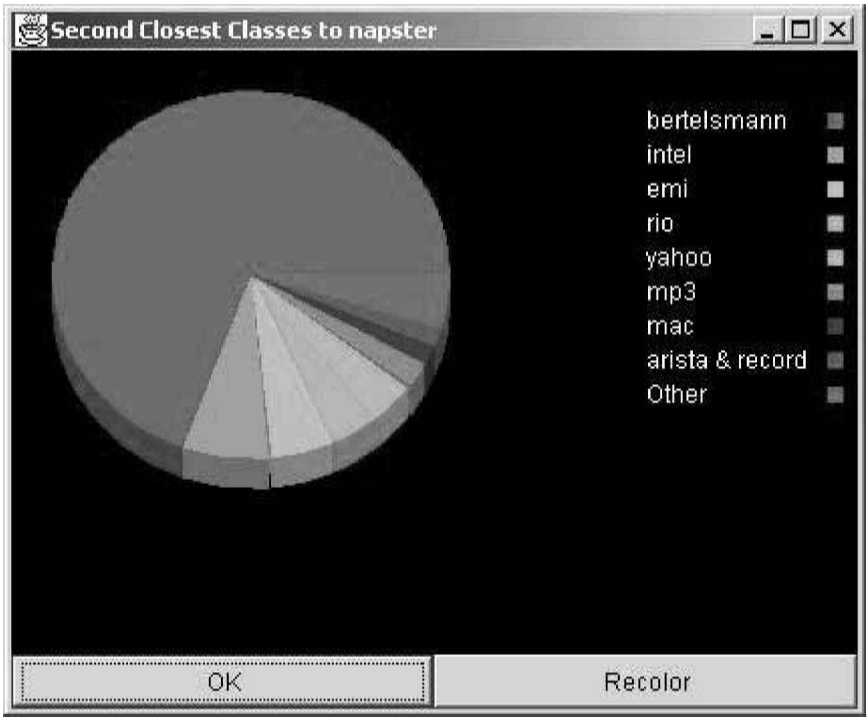


Figure 4. Similar Categories Display

sist of moving or copying selected documents from a source category to a destination category. The most difficult part of this operation from the users point of view is selecting exactly the right set of documents to move so that the source and destination categories are changed in the manner desired. To facilitate this, eClassifier provides a number of mechanisms for selecting documents.

Selection by Key Words

One of the most natural and common ways to select a set of documents is with a key word query. eClassifier allows the user to enter a query for the whole document collection or for just a specific category. The query can contain key words separated by "and" or "or" and also negated words. Words that co-occur with the query string are displayed for the user to help refine the query. Documents that are found using the key word query tool can be immediately viewed and selected one at a time or as a group to move or establish a new category.

Selection by Sorting

Another way to select documents to move or copy is via the "most/least typical" sorting technique described in the Sorting of Examples section. For example, the

documents that are least typical of a given category can be located, selected, and moved out of the category they are in. They may then be placed elsewhere or in a new category.

Selection by Visualization

The scatter plot visualization display described in the Document Level subsection can also be a powerful tool for selecting individual or groups of documents. Using a "floating box," groups of contiguous points (documents) can be selected and moved to the new desired category.

To Move, Copy, or Delete . . .

Independent of the document selection method, the user is allowed to choose between moving, copying, or deleting the selected documents. Moving is generally preferable because single-category membership generally leads to more distinct categories, which are better for the classification of future documents. Still, in cases where a more ambiguous category membership better reflects the user's natural understanding of the taxonomy, eClassifier allows the user to create a copy of the documents to be moved and to place this copy in the destination category. In such cases, the individual document will actually exist in two (or more) categories at once, until or unless the user deletes the example. Deletion is the third option. It allows the document to be removed entirely from the taxonomy, if it is judged to be not applicable.

Validation

WHENEVER A CHANGE IS MADE TO THE TAXONOMY, it is very important for the user 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. eClassifier contains a number of capabilities that allow the user to inspect the results of modifications. The goal is to ensure both that the categories are all 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 in the second section provides many 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 an especially valuable way to quickly ascertain that a category does not contain any 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.

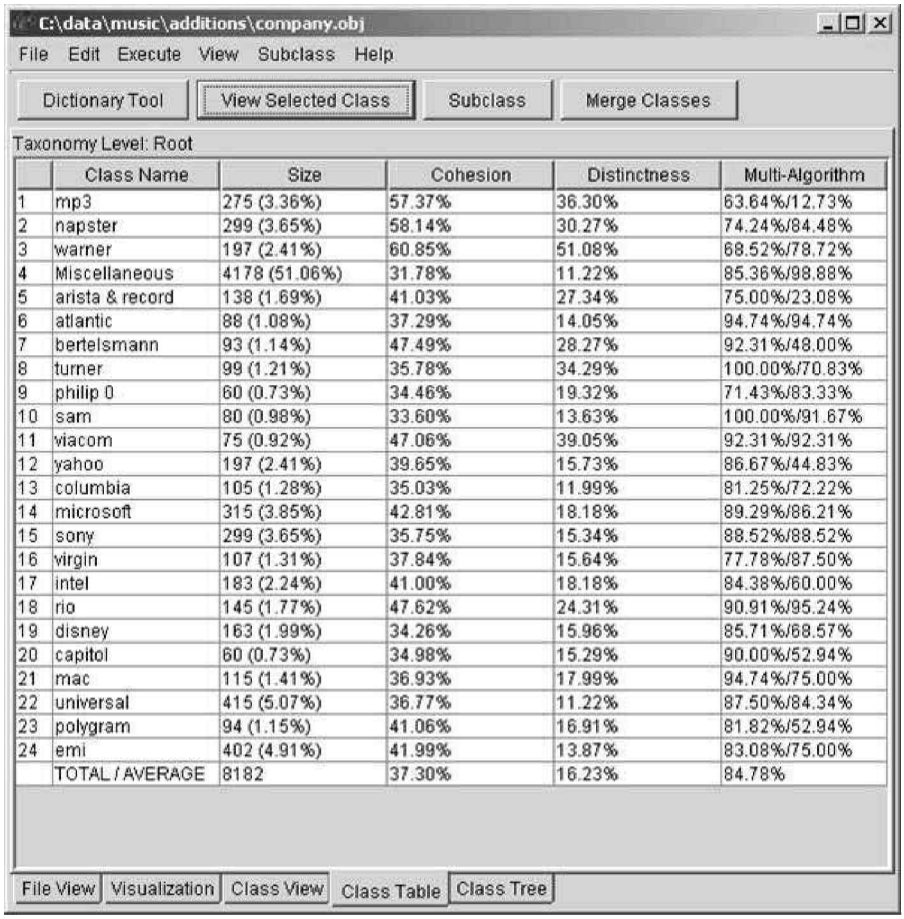
Validation Metrics

Much research has been done in the area of evaluating the results of clustering algorithms [10, 20]. Whereas 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 [21, 22], whereas admittedly a very coarse reflection of the documents actual content, does at least allow us to summarize a singe level of the taxonomy via some useful statistics. These include:

- Cohesion—a measure of similarity within category. This is the average cosine distance of the documents within a category to the centroid of that category.
- Distinctness—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 to the ones proposed by [1]: compactness and separation. The advantage of using this approach as opposed to other statistical validation techniques is that they 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 "miscellaneous" categories. These categories have a diffuse population of documents that talk about many different things. Such categories may need to be split into one or more additional categories. The second potential problem is when two different categories have very similar content. If two or more categories are almost indistinguishable in terms of their word content, they may be candidates for merging into a single category.

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 category that the user has created is not cohesive, then there is some doubt as to whether a classifier could learn to recognize a new document as belonging to that category, because the category is not well-defined in terms of word content. 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 a classifier may have difficulty distinguishing which of the two similar categories to place a candidate document in. Of course, cohesion and distinctness are rough and relative metrics, therefore 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 to the user that the cohesion and distinctness score for the new category be no worse than the average for the current level of the taxonomy (see Figure 5).



	Class Name	Size	Cohesion	Distinctness	Multi-Algorithm
1	mp3	275 (3.36%)	57.37%	36.30%	63.64%/12.73%
2	napster	299 (3.65%)	58.14%	30.27%	74.24%/84.48%
3	warner	197 (2.41%)	60.85%	51.08%	68.52%/78.72%
4	Miscellaneous	4178 (51.06%)	31.78%	11.22%	85.36%/98.88%
5	arista & record	138 (1.69%)	41.03%	27.34%	75.00%/23.08%
6	atlantic	88 (1.08%)	37.29%	14.05%	94.74%/94.74%
7	bertelsmann	93 (1.14%)	47.49%	28.27%	92.31%/48.00%
8	turner	99 (1.21%)	35.78%	34.29%	100.00%/70.83%
9	philip 0	60 (0.73%)	34.46%	19.32%	71.43%/83.33%
10	sam	80 (0.98%)	33.60%	13.63%	100.00%/91.67%
11	viacom	75 (0.92%)	47.06%	39.05%	92.31%/92.31%
12	yahoo	197 (2.41%)	39.65%	15.73%	86.67%/44.83%
13	columbia	105 (1.28%)	35.03%	11.99%	81.25%/72.22%
14	microsoft	315 (3.85%)	42.81%	18.18%	89.29%/86.21%
15	sony	299 (3.65%)	35.75%	15.34%	88.52%/88.52%
16	virgin	107 (1.31%)	37.84%	15.64%	77.78%/87.50%
17	intel	183 (2.24%)	41.00%	18.18%	84.38%/60.00%
18	rio	145 (1.77%)	47.62%	24.31%	90.91%/95.24%
19	disney	163 (1.99%)	34.26%	15.96%	85.71%/68.57%
20	capitol	60 (0.73%)	34.98%	15.29%	90.00%/52.94%
21	mac	115 (1.41%)	36.93%	17.99%	94.74%/75.00%
22	universal	415 (5.07%)	36.77%	11.22%	87.50%/84.34%
23	polygram	94 (1.15%)	41.06%	16.91%	81.82%/52.94%
24	emi	402 (4.91%)	41.99%	13.87%	83.08%/75.00%
	TOTAL / AVERAGE	8182	37.30%	16.23%	84.78%

Figure 5. Validity Measures

Applying Simple Classifiers

Metrics such as cohesion and distinctness provide a rough measure of how well a given document taxonomy can be modeled and used to classify new documents. A more accurate measure can be created by applying a suite of simple classification algorithms to a training sample of the data and seeing how accurately such classifiers work on a corresponding unseen test sample. If one or more of the classifiers can achieve a high level of accuracy on each of the categories, this indicates that there is sufficient regularity in the document word content to accurately categorize new documents, assuming the right modeling approach is used.

In eClassifier, we use a three basic approaches, centroid based, naive-bayes (multivariate and multinomial) [24], and ID3 decision tree [18] to generate classifiers from a document collection. These were chosen because they are well established in the research community, easy to program, and also very quick to train and test—speed is

of the essence if we are to give the user immediate feedback on their taxonomy. Multiple approaches are necessary since taxonomies that are created and edited arbitrarily by human experts cannot always be modeled with a single approach. After modeling, a precision and recall score is displayed for each category that indicates how well that category can be modeled with that approach. Categories that cannot be modeled with any approach should be reexamined to see if they can be modified to make them more modelable.

Ultimately, once all taxonomies have been generated, edited, and validated, they are saved in a data structure. The data structure records the system or user-given category name of each category, the membership of each example, a centroid (mean) in the term occurrence vector space corresponding to each category, and a classification model for the taxonomy. In addition, the term occurrence matrix is saved, so that we have a record of what terms occur in what documents.

We would like to make one final comment on the process of taxonomy creation, editing, and validation. Clearly, the efficacy of our approach relies heavily on the user's ability to define meaningful and useful taxonomies over a data set of documents. This requires at the very least a certain degree of familiarity at a high level with the concepts discussed in the document corpus, and ideally a high degree of subject matter expertise. Ultimately, the quality and efficacy of the generated taxonomies when applied to document exploration and retrieval will depend to a large degree on the user's level of expertise in the domain of the corpus.

The Radial Graph

ONCE MULTIPLE TAXONOMIES HAVE BEEN CREATED and validated, the challenge becomes allowing users who seek information from documents to quickly find the category or categories in the various taxonomies that are most relevant to the topic they wish to investigate. For this purpose, we have developed a multiple taxonomy browser, called MindMap, that allows user to easily navigate multiple taxonomies. MindMap can work with taxonomies developed via eClassifier, or with any other method. In this section, we describe the first phase of the MindMap browsing process. MindMap presents the user with a radial graph representing eight categories, selected from among all of the taxonomies, that best match an entered query.

These categories are selected by first converting the query into the vector space model representation. The query is then compared with every category in each of the taxonomies. First, those taxonomies whose dictionaries contain the greatest number of the key words in the query are selected. From among these the eight categories whose centroids are closest to the query in the vector space model are displayed. The radial graph of these categories presented to the user has a node representing the query at its center, and the categories color-coded by taxonomy surrounding the query. The edges of the graph vary in thickness and stroke, indicating how closely associated they are to the query. The user can now select one of these categories to further explore in the binary tree view, described below, or further refine the query by selecting from a list of related terms presented on the left of the graph in Figure 6 [5].

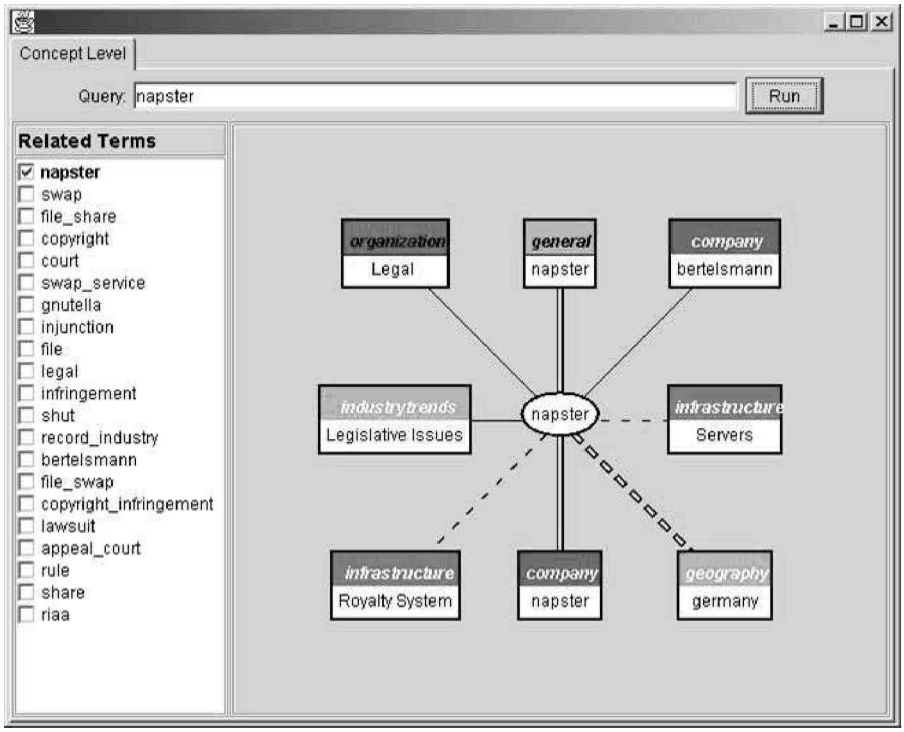


Figure 6. Radial Graph

The list of related terms is calculated by counting the co-occurrence of every word in the dictionary with the query string and then using the chi-squared test, used to determine the independence of two discrete random variables, to find the 40 most related terms (those with the lowest probability) [17]. Since the dictionaries (vector space features) may vary between taxonomies, we allow the user to select the most relevant taxonomy and use the corresponding dictionary to be for the related term calculation. Selecting any check box immediately adds the selected term to the query string. This may then cause the radial graph to change since the vector used to compare to the centroids has changed. Therefore the radial graph is updated both in terms of which nodes displayed and line thickness.

It is often the case that the user is not only interested in a category in an individual taxonomy, but the relationship between categories in different taxonomies. For example, a user may be interested in the petroleum industry in North America. The radial graph provides an easy way to explore this relationship. By taking the graph node representing the petroleum industry in the industry taxonomy and dropping it onto the node representing North America in the geography taxonomy a new node is created representing the intersection between these two categories. The graph is adjusted so that the edge between the query node and the combined category represents the strength of the relationship between the merged category and the query. The new node can now be selected in order to explore documents in the intersection of these two categories.

The radial graph provides a simple and intuitive way for users to find out which categories are relevant to the topic that they are exploring. Perhaps more important, it provides a way for users to get a sense of the relationships between taxonomies, sometimes finding surprising relationships that will aid them in gaining knowledge from the documents. The list of related terms allow users to refine their query in a sensible manner, and suggests themes to explore that may not have been otherwise known or considered. By combining nodes the user can quickly explore the relationship between the concepts represented by different taxonomies. An example is shown in Figure 6.

The Binary Tree

ONCE THE USER SELECTS A CATEGORY from the radial graph, a binary tree is presented that can be used to further refine the query. The root node of the tree represents the entire collection of documents matching the user's query in the selected category. Each branching of the tree divides the documents based on whether or not they contain some word (i.e., kd-tree) [9]. The tree is initially expanded three levels, with each branching based on the word that most evenly divides the documents represented by the parent node. An example is shown in Figure 7.

Each node displays the number and percentage of documents represented by the node, and the word whose presence or absence characterizes it. A user can select to change the word to split on at any level in the tree, as well as contract previous expansions. When the user decides to split a node, a list of the five dictionary words that most evenly split the documents are presented, but a user can also select to choose from all of the words in the dictionary. If the user does not explicitly select a word to split on, previously unexpanded nodes are split on the word that most evenly divides the documents. When the user has found a subset of documents to examine more closely, the documents can be viewed in the category visualization screen described in the seventh section.

The advantage of this approach to query refinement is that it allows the user to narrow the investigation to a manageable subset of documents without having to think of all the right words up front. The user can also gain insight into the structure of the category, quickly discovering important subdivisions.

Category Visualization

AFTER THE USER SELECTS A NODE from the MindMap binary tree, the next step is to present to the user those examples that match the query. One simple way to do this would be to merely list the examples in order of increasing distance from the centroid of the category. The disadvantage of this approach is that it essentially reduces our understanding of each example to a single number. More detailed statistics about each document are available through the word occurrence matrix calculated earlier. We can use this information to represent each document as a point in a high-dimensional geometric space, the position being related to the words the document contains [19].

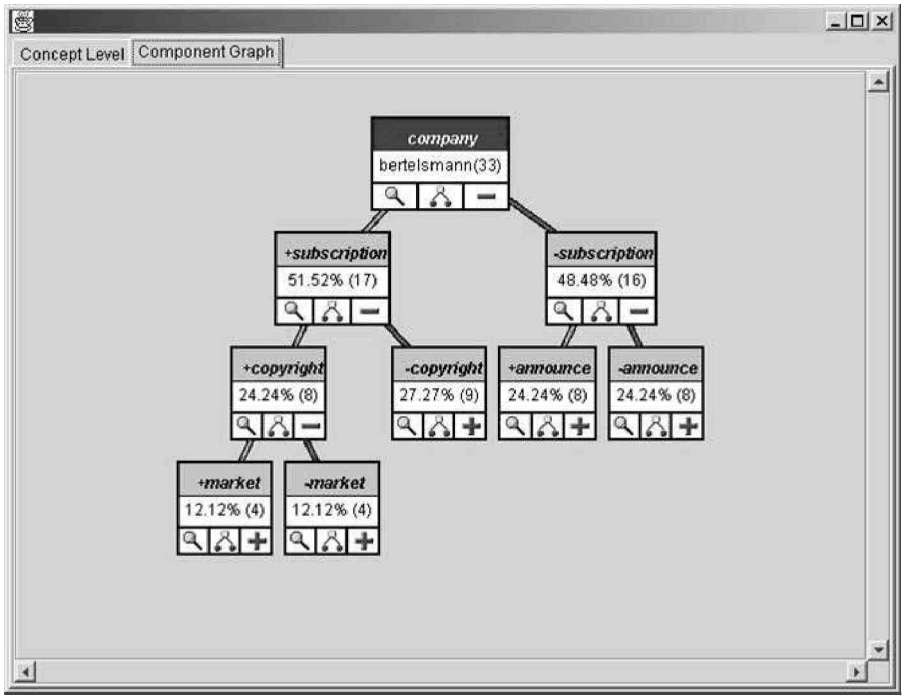


Figure 7. Napster Binary Tree

The high dimensionality (one dimension for each term) of this space makes it difficult to visualize in two dimensions. To get around this problem we use the same approach described in the Document Level subsection, finding an "interesting" plane in the high-dimensional space and projecting onto this plane by finding the intersection between the plane and a normal line drawn to each point. The most interesting plane from the perspective of distinguishing categories of examples is a plane drawn through the centroid of three categories [6]. It is important to establish a coordinate system via *points of interest* on the display, in order to make the visualization meaningful [16]. We do this by labeling the centroid of each cluster at the position of its projected spatial coordinate.

In MindMap, we display the category chosen by the user along with two other categories that are nearest neighbors to the chosen category determined using the cosine distance between centroids. The centroids of these three categories define the plane of the visualization, and all points in the three categories that match the query are displayed in the visualization. An example of such a plot is shown in Figure 8.

The points are colored based on category membership. Each point may be investigated further by placing the mouse over it to see a short document excerpt or clicking on the point to view the document text.

The advantage of this visualization is that documents that are near to each other in space should also share many terms. This allows the user to quickly locate documents related to a document of interest. In addition, by showing categories related to the

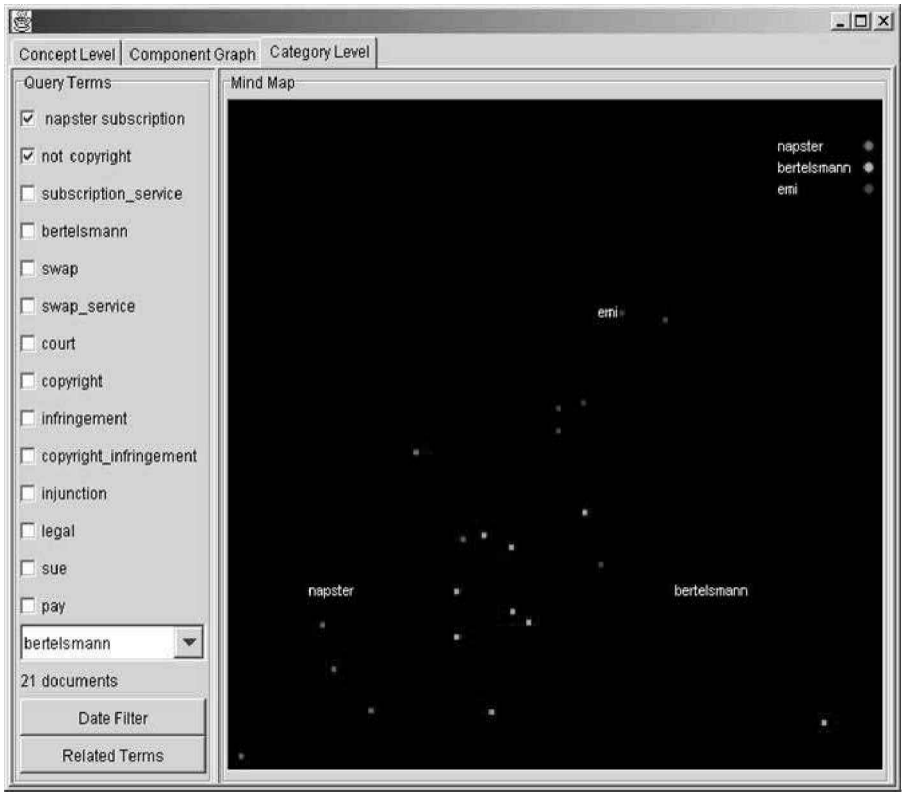


Figure 8. Napster Document Visualization

category the user has selected, we may find documents just outside the users area of interest that are related, and perhaps relevant to, the users query. One weakness of the current visualization approach is that it focuses on only one taxonomy at a time. A possible improvement would be to show similar displays of some related taxonomies in additional panels.

Tests of the MindMap prototype with typical users revealed the necessity to simplify the visualization plots as much as possible. Hence, we only display three categories at a time. Further simplifications may be necessary depending on the sophistication of the user community (e.g., displaying only two categories at a time and using the origin as the third centroid, or even hiding all points not in the selected user category). Alternative methods of reducing the dimensionality of the feature space, such as self-organizing maps [3], may also prove useful.

User Scenario

THE FOLLOWING PICTURES REPRESENT a typical usage of MindMap on a data set of music articles downloaded from the Web. We assume our user is a music company

executive who is interested in determining what other music companies are strategically aligned with Napster. The investigation begins by entering the key word "Napster" as an initial query. MindMap displays the diagram shown in Figure 6.

Notice that in addition to the company "Napster" the company "Bertelsmann" comes up as a related concept. The related words that are presented help the user to refine the query to a specific area of interest. Notice that in addition to single words, commonly co-occurring phrases are also displayed in the related terms list. Next, our hypothetical user selects the "Bertlesmann" category because this is a company the user knows has a strategic relationship with Napster. Keep in mind that the selected category represents those documents that are focused on or talk about the Bertelsmann company, not just those that may contain the work "Bertelsmann." Selecting this node brings up a further breakdown of the Bertelsmann category by system-selected dictionary terms. The result is shown in Figure 7.

Let us say that the user is interested in the subscription issue, but not in copyrights, so that therefore the user selects the "-copyright" node underneath the "+subscription" node of the tree (the node containing nine examples). This brings up the category visualization screen shown in Figure 8. Note that more than nine document icons are displayed because matching documents in two neighboring categories are displayed as well.

The two categories that are nearest to Bertelsmann are displayed along with it and only those documents containing the text "Napster" and "subscription" and not containing "copyright" are displayed. In Figure 8 we see that Napster and EMI were selected by the system because of their dictionary vector space proximity. Although the Bertelsmann relationship with Napster is well-known to our user, the relationship between Napster, Bertelsmann, and EMI is less so. Thus, MindMap has revealed a high-level relationship at the macro level that can be communicated to the user independent of any specific document. To find out more detailed information, the user selects an EMI document near the "Bertelsmann" concept, resulting in the display shown in Figure 9.

Thus, the user has rapidly found a document discussing strategic relationships in the music industry related to Napster and subscription services. Formulating a query that would discover this same document without pulling down a thousand other uninteresting documents would not be nearly as easy for most users. Clearly, in some instances the MindMap approach allows the user to find a specific type of document much more readily than standard key word queries alone can.

Scalability

THE EXAMPLE DOCUMENT SETS THAT WE HAVE TESTED using this approach have ranged in size from 5,000 to 30,000 documents. The total size of the largest text corpus processed is 100Mb. The text corpus is represented in memory as a sparse term occurrence matrix. This matrix is stored on disk for each classification until needed, and then loaded into memory on demand. This allows us to work with a very large number of different classifications, in a relatively small memory footprint. We also keep

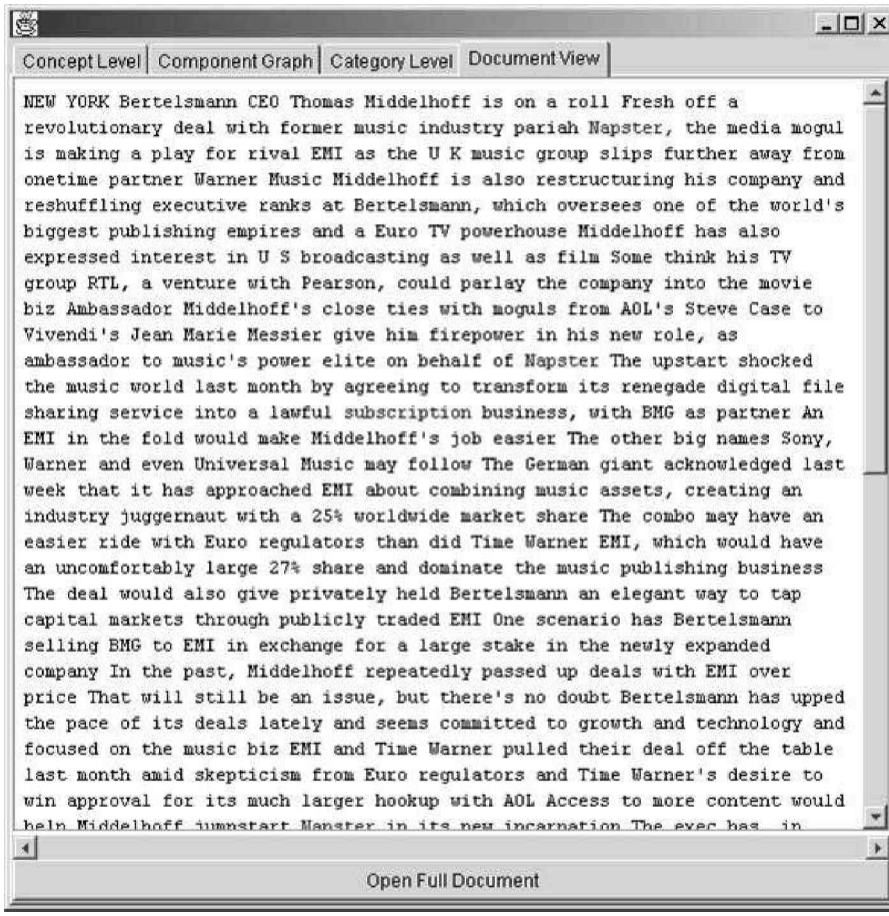


Figure 9. Napster-Related Document

in memory a representation of each classification containing, including that classifications dictionary (usually around 2,000 terms) along with the centroid vector for each cluster in each classification. The centroid is a dense vector containing a floating-point number for each dictionary term. Most of our implementations have used less than 10 classifications, each containing between 10 and 20 clusters.

If the dictionary size and number of classifications remain constant, tests show this approach should scale up to about 1–2 GB of text information on low-end PC hardware (256Mb of RAM and 500Mhz-clock speed). The need for RAM increases linearly with an increase in dictionary size or with an increase in the total number of clusters. The scalability issues would probably preclude our MindMap implementation from being used in conjunction with a general search engine generating queries over the entire World Wide Web. We feel this approach is more suitable for much small to medium-sized document collections, such as abstracts for corporate strategic document repositories.

Conclusion and Future Directions

IN SUMMARY, WE HAVE DESCRIBED A SYSTEM and methodology for the exploration of topics and concepts contained in a document collection. Using eClassifier, we have generated multiple taxonomies over documents by providing advanced visualization, editing, and validation tools to a domain expert. We then leveraged these multiple taxonomies along with related terms, visualization, and user interaction to allow for a comprehensive and flexible investigation of the content using the MindMap interface.

The advantage of our approach is that it gives the expert user an explicit way to capture this expertise and to share it throughout the information organization. Unfortunately, this user-directed approach is inherently difficult to evaluate, except in an anecdotal fashion. In addition, standard measures such as precision and accuracy of retrieval do not directly apply during the document navigation exercise since the actual endpoint or result of the MindMap visualization is not a specific set of documents, but a continuous exploration of the document space. Thus, we leave ourselves open to the criticism that our method is an ad hoc collection of techniques whose efficacy cannot be rigorously tested. Whereas the authors do not dispute the desirability of such rigorous testing when it is feasible (indeed, we use precision and accuracy during the taxonomy validation phase in a rigorous fashion), we also feel that it should not limit the development, integration, and application of proven techniques and algorithms in this space. Our contribution is to show how these multiple text mining techniques and approaches can be merged together into a coherent taxonomy generation and editing tool and to demonstrate the power of this approach when applied to browsing document collections. To date, much research has been spent on finding the "best" single taxonomy for a given document collection.

We believe there is much more that can be done to enable more understanding and exploration within document collections. User studies need to take place to determine the efficacy of each of the methods described here. There is a potential for many types of analytics that provide the user with deeper insights into concepts and relationships. We believe that techniques for automatically finding relationships across multiple taxonomies will be very promising. Incorporation of ontological technologies would help provide a more semantic interpretation of a broader set of vocabulary usage [15]. We believe that techniques borrowed from the information extraction research community will provide additional insights and enable deeper analytics. Moreover, as technology continues to make better sense of the unstructured information contained in documents, it will be compelling to relate this information to more traditional structured information, such as that found in business intelligence data warehouses.

Acknowledgments: The authors gratefully acknowledge Dharmendra Modha and Ray Strong for their contributions to the original design of eClassifier; Larry Proctor for initiating the MindMap project; and Amy Chow, Michael Danke, and JP Pietrzak for providing feedback on the user interface.

References

1. Berry, J., and Linoff, G. *Data Mining Techniques for Marketing, Sales, and Customer Support.* New York: John Wiley & Sons, 1997.

2. Brachman, R., and Anand, T. The process of knowledge discovery in databases. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.), *Advances in Knowledge Discovery and Data Mining.* Cambridge, MA: AAAI/MIT Press, 1996, pp. 37–58.

3. Chen, H.; Schuffels C.; and Orwig, R. Internet categorization and search: A self-organizing approach. *Journal of Visual Communications and Image Representation, 7,* 1 (1996), 88–102.

4. Chen, H.; Hsu, P.; Orwig, R.; Hoopes, L.; and Nunamaker, J.F., Jr. Automatic concept classification of text from electronic meetings. *Communications of the ACM, 37,* 10 (1994), 56–73.

5. Cooper, J., and Byrd, R. Lexical navigation: Visually prompted query expansion and refinement. In R.B. Allen and E. Rasmussen, *Proceedings of the Second ACM International Conference on Digital Libraries.* New York: ACM Press, 1997, pp. 237–246.

6. Dhillon, I.; Modha, D.; and Spangler, S. Visualizing class structures of multi-dimensional data. In *Proceedings of the Thirtieth Symposium on Interface: Computer Science and Statistics.* Arlington, VA: ASA, 1998, pp. 488–493 (available at www.cs.utexas.edu/users/inderjit/interface98-color.psgz).

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

8. Futakata, A. Self-organization of digital documents based on process-oriented relations. In R.H. Sprague Jr. (ed.), *Proceedings of the Thirty-Third Hawaii International Conference on System Sciences.* Los Alamitos, CA: IEEE Computer Society Press, 2000 (available at computer.org/proceedings/hicss/0493/0493toc.htm).

9. Gaede, V., and Günther, O. Multidimensional access methods. *ACM Computing Surveys, 30,* 2 (June 1998), 170–231.

10. Halkidi, M.; Batistakis, Y.; and Vazirgiannis, M. On clustering validation techniques. *Journal of Intelligent Information Systems, 17,* 2–3 (2001) 107–145.

11. Hartigan, J.A. *Clustering Algorithms.* New York: John Wiley & Sons, 1975.

12. Jing, H.; Barzilay, R.; McKeown, K.; and Elhadad, M. Summarization evaluation methods experiments and analysis. In D.R. Radev (ed.), *AAAI Intelligent Text Summarization Workshop.* Cambridge, MA: AAAI Press, 1998, pp. 60–68.

13. Jones, S.L., and Paynter, G. Interactive document summarization using automatically extracted keyphrases. In R.H. Sprague Jr. (ed.), *Proceedings of the Thirty-Fifth Hawaii International Conference on System Sciences.* Los Alamitos, CA: IEEE Computer Society Press, 2002 (available at computer.org/proceedings/hicss/1435/1435toc.htm).

14. Karjalainen, A.; Paivarinta, T.; Tyrvainen, P.; and Rajala J. Genre-based metadata for enterprise document management. In R.H. Sprague Jr. (ed.), *Proceedings of the Thirty-Third Hawaii International Conference on System Sciences.* Los Alamitos, CA: IEEE Computer Society Press, 2000 (available at computer.org/proceedings/hicss/0493/0493toc.htm).

15. Leroy, G.; Tolle, K.; and Chen, H. Customizable and ontology-enhanced medical information retrieval interfaces. In C.G. Chute (ed.), *IMIA WG6 Triennial Conference on Natural Language and Medical Concept Representation,* 1999 (available at uaeller.eller.arizona.edu/~gleroy/ais/Gondy-Leroy.htm).

16. Olsen, K.A.; Korfhage, R.R.; Sochats, K.M.; Spring, M.B.; and Williams, J.G. Visualization of a document collection: The VIBE system. *Information Processing & Management, 29,* 1 (1993), 69–81.

17. Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; and Flannery, B.P. *Numerical Recipes in C,* 2d ed. New York: Cambridge University Press, 1992.

18. Quinlan, J.R. Induction of decision trees. *Machine Learning, 1* (1987), 81–106.

19. Raghaven, V., and Wong, S. A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science, 37,* 5 (1986), 279–287.

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

21. Salton, G., and Buckley, C. Term-weighting approaches in automatic text retrieval. *Information Processing & Management, 4,* 5 (1998), 512–523.

22. Salton, G., and McGill, M.J. *Introduction to Modern Retrieval.* New York: McGraw-Hill, 1983, 52–117.

23. Vaithyanathan, S., and Dom, B. Model selection in unsupervised learning with applications to document clustering. In I. Bratko and S. S. Dzeroski (eds.), *The Sixteenth International Conference on Machine Learning.* Burlington, MA: Morgan Kaufmann, 1999, pp. 433–443.

24. Witten, I.H., and Frank, E. *Data Mining: Practical Tools and Techniques with Java Implementations.* San Francisco: Morgan Kaufmann, 2000, pp. 82–89.