SONIA: A Service for Organizing Networked Information Autonomously

Mehran Sahami[*] **Salim Yusufali**[*] **Michelle Q. W. Baldonado**[†]

[*]Gates Building 1A
Computer Science Department
Stanford University
Stanford, CA 94305-9010
{sahami, yusufali}@cs.stanford.edu

[†]Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
baldonado@parc.xerox.com

ABSTRACT

The recent explosion of on-line information in Digital Libraries and on the World Wide Web has given rise to a number of query-based search engines and manually constructed topical hierarchies. However, these tools are quickly becoming inadequate as query results grow incomprehensibly large and manual classification in topic hierarchies creates an immense bottleneck. We address these problems with a system for topical information space navigation that combines the query-based and taxonomic systems. We employ machine learning techniques to create dynamic document categorizations based on the full-text of articles that are retrieved in response to users' queries. Our system, named SONIA (*S*ervice for *O*rganizing *N*etworked *I*nformation *A*utonomously), has been implemented as part of the Stanford Digital Libraries Testbed. It employs a combination of technologies that takes the results of queries to networked information sources and, in real-time, automatically retrieve, parse and organize these documents into coherent categories for presentation to the user. Moreover, the system can then save such document organizations in user profiles which can then be used to help classify future query results by the same user. SONIA uses a multi-tier approach to extracting relevant terms from documents as well as statistical clustering methods to determine potential topics within a document collection. It also makes use of Bayesian classification techniques to classify new documents within an existing categorization scheme. In this way, it allows users to navigate the results of a query at a more topical level rather than having to examine each document text separately.

KEYWORDS: Clustering, Classification, Feature Selection, Distributed Information

INTRODUCTION

The enormous amount of information available on the World Wide Web and other networked information sources such as Digital Libraries has created an urgently pressing need to provide users with tools to navigate these information spaces. Initial attempts at addressing this problem have led to the development of a number of information finding tools such as Web-based search engines (e.g., *Alta Vista*) that allow users to specify queries that are then matched against a database of previously indexed documents. Given the enormous growth of networked information, however, the results of many queries often yield unwieldy lists of documents that flood the user with too much information, most of which is really irrelevant to their *information need*.

Alternatively, directory services (e.g., *Yahoo!*) provide users with manually constructed topic hierarchies so as to impose some higher-level navigational structure on a corpus of information. Unfortunately, such topical hierarchies currently require documents to be manually classified into appropriate topics and thus create an immense information bottleneck. Consequently, only an extremely small portion of the entire information space is captured within such a hierarchy. Also worth noting is the fact that networked information can often come from a number of heterogeneous sources (i.e., the World Wide Web, different Digital Libraries, proprietary databases, etc.), whereas many existing information finding tools are only implemented to work with one information source.

We seek to address these problems with a system for *topical* information space navigation that combines both the query-based and taxonomic approaches. Our system, named SONIA (*S*ervice for *O*rganizing *N*etworked *I*nformation *A*utonomously), employs a number of machine learning techniques, such as feature selection, clustering, and classification, to create dynamic document categorizations based on the full-text of articles that are retrieved in response to users' queries. In this way, users can explicitly specify their information needs as queries while also having the ability to browse the results of their queries at a topical, rather than document, level.

Related work in this area, most notably the Scatter/Gather approach [6], has shown that document clustering is an effective way for allowing users to quickly hone in on the documents relevant to them. Moreover, document clustering can also be useful for navigating query results [10], and specialized user interfaces have been developed for such document clustering systems. For example, Allen, Obry and Littman [2] have developed an interface that allows users to navigate through the *dendogram* of documents generated by a hierarchical agglomerative clustering algorithm. In this way, users can potentially locate subsets of particularly relevant documents.

Other researchers have focused on the use of visualization methods for conveying similarity between documents to the user. Such systems rely on the Cluster Hypothesis which states that "closely associated documents tend to be relevant to the same requests" [22]. Accordingly, the system of Allan, Leouski, and Swan [1] conveys document similarity to the user via spatial layout. Here, relevant documents are often located near each other spatially. Thus, when a user locates a relevant document, it is more likely that they will find other relevant documents by examining the local neighborhood.

Further support for the Cluster Hypothesis comes from the empirical observation that clustering tends to concentrate documents particularly relevant to a query in just one or two groupings [11]. Moreover, this work has shown that users are generally successful at locating a higher proportion of relevant documents by simply identifying the appropriate high-level groupings.

Another example of the use of clustering to aid in information access includes the WEBSOM system [21]. WEBSOM uses a Self-Organizing Map (SOM) [12] to group together related words into a *word category map*. This map is in turn used to automatically organize documents according to the words that they contain. The interface to this system then allows users to navigate the word category map and zoom in on groups of documents related to a particular group of words. This approach seems to require a fairly sizable initial corpus to generate a useful word category map and thus may not be directly applicable to query results.

While our system embodies several similar elements to those in previous work, it uses entirely different technologies to realize this functionality. More importantly, however, SONIA provides significant new extensions in terms of technical functionality and broader applicability. Operating in the dynamic context of networked information, SONIA makes use of a number of methods for relevant feature extraction from documents through a multi-tiered feature selection process that is customized to each user query. Furthermore, since our system exists as part of a general architecture within the Stanford Digital Libraries Testbed [9], it has the ability to simultaneously retrieve information from a number of heterogeneous sources, thereby making our system maximally flexible. SONIA was also designed with efficiency in mind, thereby facilitating real-time user interactivity even when accessing diverse, distributed document collections.

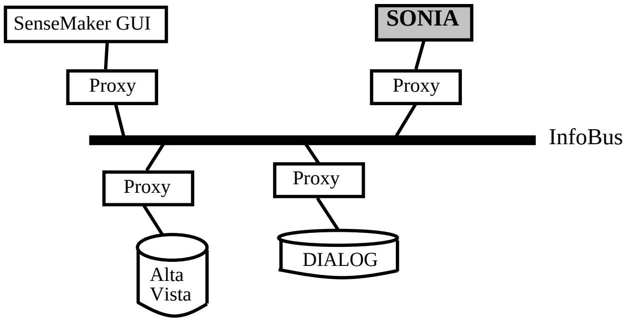

Figure 1: The InfoBus architecture.

The most significant extension of SONIA beyond existing systems, however, is the ability to save various document clusterings (i.e. topical partitionings) as classification schemes that can be used to automatically categorize the results of subsequent, but related, queries. This combination of clustering and classification allows users to not only navigate a given document collection more easily, but enables them to quickly construct and maintain their own organizational structures for the vast quantities of information available to them. In this way, we hope to elevate user interaction with Digital Libraries beyond simple one-shot queries and move to addressing users' more persistent information needs.

In the remainder of this paper we present the technical details of SONIA and the architecture in which it is embedded. We also provide a detailed account of the machine learning methods employed in SONIA and provide empirical results with these methods in several controlled experiments. We then show examples of the system in use, discussing its efficacy in information browsing and classification. Finally, we give a summary of this work and its future directions.

SYSTEM OVERVIEW

To get a complete picture of how SONIA is used, it becomes necessary to first understand the architecture in which it exists. Thus, we presently give a brief description of the Stanford Digital Libraries InfoBus Architecture, showing how SONIA is situated within a larger distributed systems context. Subsequently, we give a detailed description of the components that comprise SONIA.

InfoBus Architecture

The focus of the Stanford Digital Libraries project is on providing interoperability among heterogeneous, distributed information sources, services and interfaces. To this end, the InfoBus architecture [3] shown in Figure 1 has been developed. In brief, the InfoBus is comprised of network proxies that encapsulate the protocols used by disparate interfaces, information sources, and information services. These proxies allow for communication among the different entities connected to the InfoBus by translating their communications into a common language.

SONIA exists within this architecture as an information service with a number of capabilities. First, it allows for the clustering of collections of documents to help extract *topical* descriptions. This allows users to more quickly find sub-collections of documents that satisfy their information needs, and thus ignore much of the irrelevant material often returned by simple queries. Furthermore, SONIA also allows for such document groupings to be stored as persistent categorization schemes (referred to as *profiles*). Each profile is simply a partitioning of documents into a number of semantically meaningful groups. In this way, new query results can be integrated into a topical partitioning derived from previous query results. This allows the user to build up a large collection of results spanning multiple related queries within the same organizational scheme.

Currently, SONIA is accessed through the Java-based *SenseMaker* interface [4], which allows users to simultaneously query multiple heterogeneous information sources including popular Web search engines, proprietary information databases (e.g., DIALOG) and many others. SenseMaker can then be used to organize documents by matching titles, matching URL's (for Web documents), and the like, or it can utilize SONIA to group documents by their full-text content. At this point, a user can either specify that a set of documents should be grouped in accordance with a previously saved profile (categorization scheme) or, if no existing profile is used, that the documents should be clustered into a new categorization scheme. A user can choose to save any such categorization as a persistent profile for future use, or update an existing profile with additional documents that are classified into it. Moreover, SONIA allows a single user to have several distinct profiles to reflect each of his or her diverse information needs. To better understand the technologies incorporated within the system, we presently turn our attention to the components that comprise SONIA.

SONIA

It is simplest to view SONIA as a series of modules, each of which is responsible for a data transformation procedure. Figure 2 presents an overview of these modules.

Document retrieval and parsing Since on-line information sources are rapidly changing, SONIA does not attempt to maintain its own (possibly outdated) inverted index of documents, but rather treats networked information as a massive digital library from which it can dynamically retrieve documents. As a result, SONIA is given as input only a list of document identifiers (e.g, URL's for Web documents, ID numbers for DIALOG, etc.) and then employs a highly parallelized document retrieval module (sometimes called a network *crawler* or *spider*) to retrieve the full text of the corresponding documents. This module does not present a timing bottleneck in real-time interaction as it is capable of robustly retrieving as many as 250 document texts in parallel, and utilizes a time-out condition to prevent needlessly long waits for documents.

The retrieved document texts are then parsed into a series of alphanumeric terms (i.e., words). Optionally, these terms may be stemmed to their root as SONIA's parser includes a standard word stemming scheme [17]. Each term then forms a dimension in a high-dimensional vector-space in which the documents can now be represented as points. That is, the vector representing a document contains in the dimension for each term, the count of how many times that term appeared in the document. Since we now have the term counts for each document, SONIA is capable of transforming the vector representation of documents to different weighting schemes, such as TFIDF weights [20] or a simple Boolean representation, indicating only term appearance or non-appearance in documents. Such different representations are easily generated when needed by different modules within SONIA.

Multi-tiered feature selection Since the number of distinct terms in unrestricted text is very large (i.e., some small collections have exhibited as many as 10^5 distinct terms), feature selection becomes necessary. SONIA uses a multi-tier feature selection process, using both Natural Language phenomena as well as statistical machine learning techniques to reduce the feature space drastically. The system incorporates four forms of feature selection, each of which operates on the vector-space representation of the documents. Initially, dimensions representing *stop words* (non-meaningful terms, such as "a" and "the") are eliminated from the document vectors. These stop words are determined using a standard English stop word list of 570 words as well as a hand-crafted list of approximately 100 Web stop words (such as "html" and "url").

In the second-tier of feature selection, a Zipf's Law analysis [22] of term occurrence over the collection is used. This essentially eliminates terms that appear fewer than 3 or greater than 1000 times in the entire collection as not having adequate resolving power to differentiate sub-collections of documents. These threshold values were chosen since they appear to work quite well in practice.

After these first two stages of feature selection, the system reaches a branching point depending on the user's choice to organize the current set of documents with respect to an existing profile or not. If a profile is being used, then we are working in the context of a *supervised* learning problem in which case we can make use of information in the existing profile to classify documents accordingly. If a profile is not being employed, then we must create a document organization from scratch and are thus working in the context of *unsupervised* learning.

We first consider the case where an existing profile is being employed. Here, a previous grouping of documents has been stored in a *Profile Database* maintained by SONIA. Since

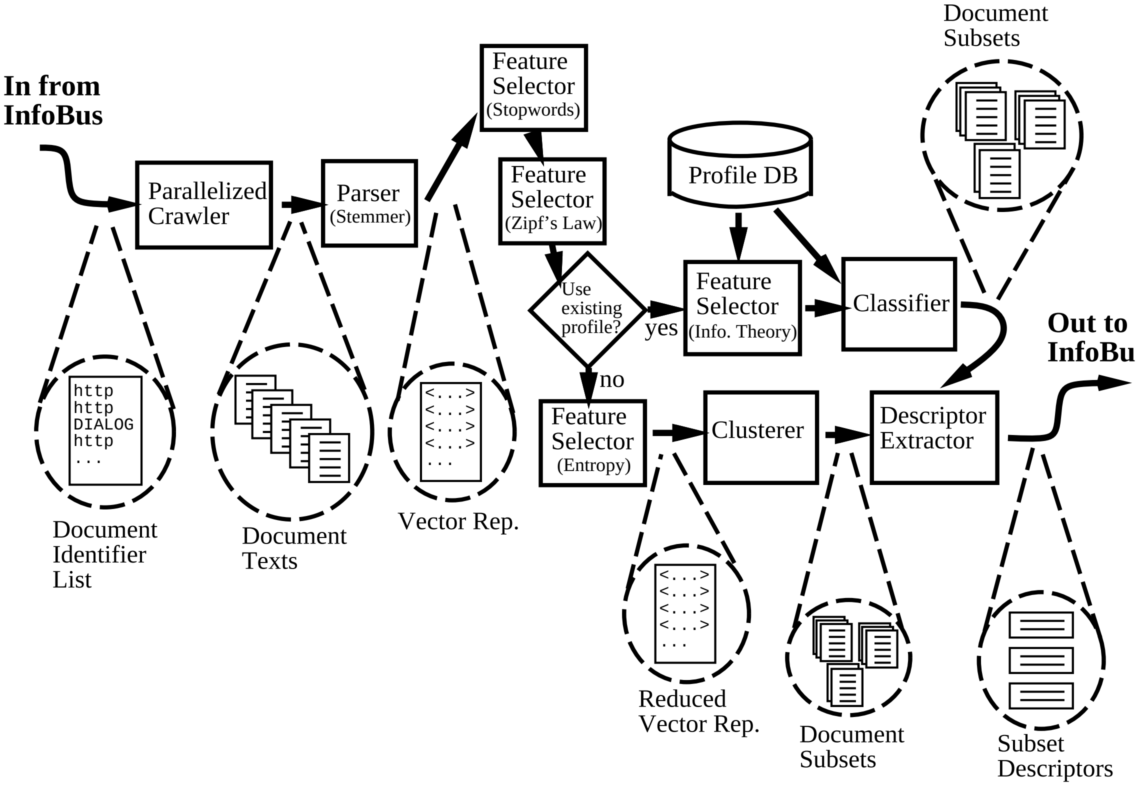

Figure 2: Processing stages in the SONIA system.

we wish to organize the current set of documents according to this previous classification scheme, we need to find the terms that are most discriminating between groups in the given profile. To this end, we employ a form of information theoretic feature selection that is effective for a number of similar classification problems, including text categorization [13]. Moreover, we have also found in previous work [14] that very few terms are needed for accurate document classification. We corroborate these early findings with further experimental results presented later in this paper, showing that 50 features are quite sufficient for accurate classification. Thus, we aggressively reduce the feature set at this point, from several thousand terms to the just the 50 most discriminating ones.

In the case where the user chooses not to categorize documents according to an existing profile, but instead wishes that a new categorization be created, we consider a different form of feature selection based on an entropy criterion [5]. Here we hone in on terms with high distributional variability among documents, making them likely to identify subtopics within a varied collection. For each term t_i we compute the probability of its occurrence in a randomly chosen document from our collection. Thus, we define $P(t_i) = \frac{|D_{t_i}|}{|D|}$ where D is the total collection of documents and D_{t_i} is the subset of D comprised of only those documents that contain term t_i. We can now compute the entropy, H, of term t_i as $H(t_i) = -P(t_i)\log_2 P(t_i)$. We use this entropy metric to eliminate those terms with the *least* entropy since we wish to retain terms with highly varied distributions. This reduction is currently applied to eliminate 15% of the terms remaining after the first two stages of feature selection.

Note that we are not as aggressive in performing feature selection here as in the case where a profile is used. The reason for this is that we have no direct objective function tying the features selected here to the subsequent discovery of a good organizational scheme (as we do when a profile is present). Thus, we choose to be conservative by keeping more terms. Moreover, the clustering algorithms we employ are less computationally intensive than those used for classification. Hence the fact that we keep more features in this case is not as serious a hindrance.

Classification If the user chooses to categorize documents using an existing profile, then we simply have a traditional machine learning classification problem on our hands. Here, the documents in the existing profile become the training set and the partitioning defined by the profile defines the classes in the data. From this data, a classifier is built that can then be used to classify incoming documents.

While SONIA provides full generality to use any classification algorithm, we have chosen to focus on techniques based on Bayesian networks. Currently, we use the Naive Bayesian classification algorithm [8]. This algorithm attempts to predict for each document, d, the category, c_j, for which it has maximal probability. Formally, this is given by

$$argmax_{c_j \in C} P(c_j|d) = argmax_{c_j \in C} \frac{P(d|c_j) \cdot P(c_j)}{P(d)} \quad (1)$$

where C denotes the set of all possible categories. Since the value of $P(d)$ is the same regardless of category, we need not compute this term explicitly in order to find the maximally probable category. Note, however, that d is a n-dimensional Boolean vector of term appearances, $t_1, t_2, \ldots, t_n$, making it intractable to compute $P(t_1, t_2, \ldots, t_n|c_j)$ directly. Rather, the Naive Bayesian classifier makes the simplifying assumption that $P(t_1, t_2, \ldots, t_n|c_j) = \prod_{i=1}^{n} P(t_i|c_j)$. This corresponds to assuming that the appearance of each term is independent of every other term given the value of the category variable C. While this assumption may seem unrealistic for text, the Naive Bayesian classifier has shown very good empirical results in text domains [16]. We provide several examples of the performance of this method in controlled experiments subsequently. Nevertheless, to relax the restrictive independence assumption we have recently implemented more expressive Bayesian classification schemes [19] in SONIA, and found them to yield even better results for document classification [14].

Once the documents are classified into groups, this grouping information is passed through the InfoBus to the SenseMaker interface. These documents are then displayed according to the categories defined in the user's profile.

Clustering Alternatively, if the user did not select a profile by which documents should be classified, SONIA will employ clustering to create a novel topical categorization of the documents. As with the classification module, any reasonable clustering method can be used at this stage. We have recently conducted comparisons with a number of different clustering algorithms including hierarchical agglomerative clustering [18] and iterative clustering methods, such as K-Means [15], using different measures of similarity between documents [7]. Currently, we have chosen to use a two-step approach to clustering. First, group-average hierarchical agglomerative clustering is used to form an initial set of clusters which is then further optimized with an iterative method. Both of these methods rely on the definition of a similarity score between pairs of documents d_i and d_j which, for generality, we will refer to as $Sim(d_i, d_j)$. The similarity score currently used in SONIA is intuitively based on the notion of expected probabilistic overlap in words between a pair of documents. Formally, this measure is defined as

$$Sim(d_i, d_j) = \sum_{w \in d_i \cap d_j} \frac{P(Y_i = w|d_i) \cdot P(Y_j = w|d_j)}{P(Y = w)} \quad (2)$$

where $P(Y_i = w|d_i)$ is the probability that a word randomly selected from document d_i is equal to w, and $P(Y = w)$ is the probability that a word randomly selected from the entire corpus is equal to w. Note that the denominator in Eq. 2 provides a *scaling* of the word space, causing overlap in rare words (i.e., words with small values for $P(Y = w)$) to contribute more to the similarity measure between two documents than a match on a more common term.

To compute the probabilities in Eq. 2, namely $P(Y_i = w|d_i)$, we use a novel normalized geometric mean (NGM) smoothing estimate. A justification of this estimate is beyond the scope of this paper (we refer the interested reader to [7] for further details), but we have found it to work quite well in practice and present a brief overview of these results shortly.

The hierarchical agglomerative clustering method proceeds by initially placing each document in a separate cluster. The similarity between each pair of clusters c and c', denoted $Sim(c, c')$, is computed and the two closest clusters are then merged. We use the *group average* variety of hierarchical clustering in which the similarity between a pair of clusters is defined as the average similarity between every pair of documents in those clusters (where one document comes from each cluster). More formally,

$$Sim(c, c') = \sum_{d_i \in c, d_j \in c'} \frac{1}{|c| \cdot |c'|} Sim(d_i, d_j). \quad (3)$$

This process of computing pair-wise cluster similarities and merging the closest two clusters is repeatedly applied, generating a dendogram structure which simply contains one cluster (encompassing all the data) at its root. By selecting an appropriate level of granularity in this dendogram, it becomes possible to generate a partitioning into as many clusters as desired. Moreover, criteria, such as a minimum number of documents per cluster, are often used to prevent outlier documents from being considered a separate cluster. In our experiments we heuristically set this minimum cluster size at 10 documents.

Once an initial set of clusters is formed in this way, an iterative refinement step is employed to further optimize the results. Here, the similarity between each document and cluster (i.e., $Sim(d, c)$) is computed and each document is assigned to the cluster to which it is closest, thus defining a new clustering. This process is repeated until convergence (i.e., no documents change clusters) or until some maximum number of iterations are performed (we used a maximum of 10 iterations).

Note that the clustering methods we employ currently require that the user specify an a priori number of clusters into which the data should be grouped. The current SenseMaker interface does not allow for this value to be easily changed by the user, so we simply clamp it at a reasonable hard-coded value, generally between 2 and 10. Currently, we are exploring an extended interface to this system which easily allows users to vary this parameter.

Data set	Number of Docs	Number of Words	Categories
D1	486	1143	Natural Gas, Soybean, Dollar
D2	466	1001	Gold, Coffee, Sugar
D3	289	552	T-Bill, Yen, Reserves
D4	467	1126	GNP, Livestock, Sugar
D5	1426	1953	Loan, Interest, Money Effects

Table 1: Data sets used in controlled experiments.

Descriptor extraction Once classification or clustering has been performed, SONIA's final module extracts *descriptors* from the document subsets so that a coherent description of the topics found in the document collection can be presented to the user through the interface communicating with SONIA. More precisely, SONIA returns a grouping of the initial document identifiers into different subsets based on the results of either clustering or classification. It also returns automatically generated topical descriptors that are extracted from each such subset of documents.

We have compared a few methods for extracting these descriptors. The first such method is a probabilistic *odds* scheme in which, for each document group c_j, we compute the probabilistic odds $O_j(t_i)$ of a term t_i appearing in a document in c_j versus appearing in a document in any other group as $O_j(t_i) = \frac{P(t_i|c_j)}{\sum_{c_k \neq c_j} P(t_i|c_k)}$. We then select some number, κ, of terms with the highest O_j values as the descriptor for document subset c_j.

Alternatively, we have also considered a simple *centroid*-based approach for descriptor extraction. Here, we simply compute the Euclidean centroid of all documents assigned to each group c_j. As before, we simply take the κ terms corresponding to the dimensions with highest value in the centroid vector as the descriptor for that group. Currently, we use $\kappa = 12$, as this value appears to achieves a good balance between brevity and descriptiveness.

In practice, we have found that the centroid-based approach appears to yield words that are more indicative of the topic of a given document subset. It should be noted, however, that part of the success of the centroid-based approach relies on the efficacy of prior stop word elimination to prevent common meaningless words from appearing in the descriptor lists, since these words will be very common and hence have high frequency counts in all document subsets. In contrast, the problem with the odds based approach is that it seems to favor very rare (and hence not particularly descriptive) terms that may appear a few times in one document subset, but not in any of the others. As a result, these terms get a much higher *odds* score than more common terms that may appear even a few times in the other document subsets.

SYSTEM EVALUATION

Having detailed the myriad components that comprise SONIA, we presently give detailed examples of the complete system in action. Since it is difficult to provide an objective measure by which to evaluate such a system as a whole, we first present the results of controlled studies in which the efficacy of the clustering and classification methods implemented in SONIA can be measured directly.

Controlled Experiments

In order to measure the performance of the clustering and classification methods implemented in SONIA, we consider a set of controlled experiments using subsets of documents from the Reuters-22173 collection of newswire articles.[1] The data sets used here are created by considering only documents with one of a particular subset of hand-labeled topics from the Reuters collection. A description of the resulting data sets (named D1 through D5) is given in Table 1. The size of these data sets is chosen to reflect the number of documents that may be generally reasonable to expect back in response to a query. Moreover, both stop word elimination and Zipf's Law feature selection are applied to these data sets and the number of words reported in Table 1 reflects the number of words in each corpus *after* such feature selection is applied.

In our first set of experiments, we seek to measure the efficacy of clustering based on the measure of probabilistic word overlap used in SONIA in comparison to other state-of-the-art document similarity measures used in the Information Retrieval community. For comparison, we consider the Cosine similarity measure used in conjunction with square root word frequency dampening (as in the Scatter/Gather system [6]) and standard TFIDF weighting [20] used widely in information retrieval systems.

Realizing that methods for evaluating clustering algorithms are not without controversy, we use the following strategy, being aware of its limitations. We use the previously labeled Reuters data and measure how well the clustering methods can recover the known label structure in the data. This reflects how well each method is able to partition the document collection into the same meaningful categories as the human labeler of the data. We fix the number of clusters to be found to be the same as the known number of categories in the data. The clustering algorithms, however, are given no information about the true category of each document. After clustering has completed, we set the predicted label for all documents in each cluster to be the true label that the majority of documents in that cluster have. Given that we have an actual and predicted label for each document, we can now simply compute the classification accuracy (how often the actual and predicted label for each document coincide). A brief review of the results of these experiments is presented in Table 2. For a more comprehensive comparison of these and other related similarity measures for clustering, we refer the reader to [7]. Here, we can clearly see that the measure of probabilistic

[1] An updated version of this data set, Reuters-21578, is now available from David Lewis (`http://www.research.att.com/~lewis`).

	Data sets				
Method	D1	D2	D3	D4	D5
SONIA	99.4%	98.5%	95.8%	95.9%	79.7%
Scatter/Gather	99.4%	97.0%	76.7%	80.9%	77.6%
TFIDF	99.4%	95.7%	92.4%	60.0%	52.7%

Table 2: Accuracy results for clustering.

overlap used in SONIA is generally more successful at partitioning the data sets into meaningful groupings than the other commonly used methods. In fact, SONIA achieves comparable or superior results to the other methods in every case, sometimes outperforming them by a considerable margin.

In the second set of controlled experiments, we consider the efficacy of the Naive Bayesian classifier used in conjunction with information theoretic feature selection. Since we seek both to measure the performance of Naive Bayes on an absolute scale, as well as the relative effects of feature selection, we run Naive Bayes several times on each data set, using a different number of features in each case. We employ 10-fold cross-validation [23] for evaluation and report both the average classification accuracy and standard deviation over these 10 runs for each entry in Table 3. We also provide the overall average over all five data sets for each feature selection regime. These results indicate that Naive Bayes is quite effective for text classification and often requires far fewer features than the entire feature set in order to produce highly accurate classifications. As a matter of fact, in many cases, aggressive feature selection can help improve the classification results (albeit modestly in some cases). This general trend is most clear in looking at the average performance over all five data sets, which indicates that using only the 50 most informative features will generally produce quite acceptable classification results. This is the number of features currently selected by the algorithm implemented in SONIA.

We next turn our attention to evaluating SONIA in terms of actual usage scenarios which make use of both clustering and classification. Presently, we describe two such scenarios.

Usage Scenario One

In the first scenario, we consider the situation in which a researcher may be looking for papers by Hector Garcia-Molina, one of the principle investigators in the Stanford Digital Libraries Project. The query "Hector Garcia-Molina" is sent through the InfoBus to the *Excite* Web search engine from an interface such as SenseMaker and 200 matching URL's are returned. These URL's are then passed (again through the InfoBus) to SONIA without specifying an existing profile for classification. The crawler in SONIA is able to retrieve 141 valid Web pages from these URL's in the allotted lookup time. These pages are then parsed (we chose not to use word stemming in these examples) and feature selection is performed. The original feature space for these document is approximately 8000 distinct terms. The multi-tiered feature selection process eliminates over 5000 of these terms.

Since no existing profile was selected for categorizing these documents, they are clustered to form a new organizational scheme. The result of this clustering (into 4 categories) is shown in Table 4, which presents the descriptors extracted for each cluster, a sampling of the document titles in that cluster, and a human generated *feasible topic* denoting the readily apparent major theme of the cluster. We note that the entire process of document retrieval, parsing, feature selection, clustering and descriptor extraction takes approximately 2.5 minutes of wall clock time on a heavily loaded Sparc Ultra 2. This makes the system quite suitable for real-time usage, considering the network delay times usually associated with document browsing on the World Wide Web.

From the results in Table 4, we can see that SONIA is effective at picking out the major themes in the given document set, especially considering that it is able to distinguish between Prof. Garcia-Molina's two major lines of research, Digital Libraries and Databases. It is even able to distinguish his colleagues in those areas, as Steven Ketchpel is one of his students working on Digital Libraries while Anthony Tomasic is a colleague working in the area of distributed databases. More surprisingly, we find a cluster of documents with a number of Spanish names as descriptors. A quick perusal of the document titles in this group reveals that these are pages written in Spanish that happen to contain the common Hispanic names "Hector", "Garcia" and "Molina". By placing these pages together, SONIA is not only able to identify major topical themes in the collection, but also helps users quickly eliminate irrelevant documents that just happen to match their query.

After forming this initial partitioning of documents, the user saves this organization as a profile (named "Hector") in which to classify subsequent related queries. As one example of this, the user issues the follow-up query "Sudarshan Chawathe", having found out that Sudarshan is one of Prof. Garcia-Molina's current students. In this case, the user may only be interested in finding out how Sudarshan's work overlaps with that of his advisor, and thus only requests the top 30 URL's from the search service. The user then requests that SONIA classify these resulting URL's according to the previously saved "Hector" profile. Here, we find that SONIA retrieves 29 valid documents and classifies 27 of them in the category pertaining to Database Research, the other two in the category on Departmental Duties. A subsequent analysis of the actual documents reveals that Sudarshan does in fact work on distributed database systems and is not a member of the Stanford Digital Libraries Project. Moreover, of the 27 documents placed in the Database Research category, 26 refer specifically to research, conferences and colleagues in the area of database systems. The remaining document refers to housing options at Stanford and does not appear to fit well into any category. The two documents placed in the Depart-

Number of	Data sets					
Features	D1	D2	D3	D4	D5	Average
20	98.3% ± 2.2%	98.7% ± 2.1%	93.9% ± 4.8%	97.0% ± 1.5%	76.0% ± 3.7%	92.8%
50	99.2% ± 1.1%	97.8% ± 2.5%	91.8% ± 3.8%	97.2% ± 2.3%	78.2% ± 2.4%	92.8%
100	99.4% ± 1.0%	97.8% ± 2.0%	90.0% ± 5.0%	96.1% ± 2.5%	78.5% ± 2.0%	92.4%
200	99.8% ± 0.7%	97.8% ± 1.8%	87.5% ± 7.0%	94.6% ± 4.0%	78.7% ± 2.2%	91.7%
400	99.6% ± 0.9%	96.7% ± 2.3%	87.1% ± 7.0%	93.3% ± 4.9%	78.4% ± 3.2%	91.0%
all	99.6% ± 0.9%	96.7% ± 3.3%	86.8% ± 6.5%	93.5% ± 5.4%	79.4% ± 3.1%	91.2%

Table 3: Accuracy results using the Naive Bayesian classifier with feature selection.

Automatically Generated Descriptors	Sample Document Titles	Feasible Topics
information, stanford, digital, university, http, library, ketchpel, user, //www, project, steven, infobus	Quarterly Report Stanford Digital Library Project Agent Projects in the Stanford Digital Library Home Page - Steven Ketchpel D-Lib, November 1995	Stanford Digital Library
database, systems, garcia, hector, molina, data, distributed, abstract, 1998, system, information, michael	The VLDB Journal, Volume 1 SIGMOD Conference 1995 DB&LP: Anthony Tomasic Technical Publications	Database Research and References
computer, area, university, science, design, systems, faculty, david, (electrical, professors, engineering, 1997	CSL 1998 EE Quals Computer Science *(departmental page)* Faculty of the Center for Telecommunications Journal of the ACM Editorial Board	Departmental and Professional Duties
de, jose, gonzalez, luis, la, carlos, garcia, francisco, juan, maria, martinez, antonio	Asociados Gran Comision Arbol de tesis dirigidas SBC Validacion de Informacion Hidrologica	Spanish Language Pages

Table 4: Sample results on the query "Hector Garcia-Molina".

mental Duties category are in fact lists of graduate students in the Computer Science Department at Stanford, and arguably appear to be classified into the most appropriate available category. In any case, the vast majority of documents are classified into the correct topic and the user can not only get an immediate sense for the type of work Sudarshan does, but can now augment this organizational profile with even more documents that are related to one of Prof. Garcia-Molina's primary research areas. In this way, users can easily maintain up-to-date document collections that are automatically topically organized.

Usage Scenario Two

Now let us consider the situation in which a middle school student is writing a report about the possibility of life on Saturn. The student begins by issuing the query "Saturn" from SenseMaker to *Excite*, which returns 150 URL's. As before, these URL's are passed to SONIA without specifying an existing organizational profile, so SONIA will form a new categorization via full-text clustering. In a total wall clock time of approximately 1 minute (again on a heavily loaded Sparc Ultra 2), SONIA retrieves and parses the 103 active documents from this set of URL's, performs three stages of feature selection, clusters the documents and returns a document organization complete with category descriptors. During this process, feature selection reduced the total feature space from over 7000 initial terms to just under 900. The resulting document categorization is described in Table 5.

We find that SONIA is able to readily distinguish those documents about the planet Saturn with those about the car company, as well as the Sega Saturn video game (although this latter topic may be of more interest to a middle schooler than writing a report on the planet). This is especially important when we note that some of the web pages of Saturn car enthusiasts have such vague titles as "Saturn Talk" and "Craig's Saturn Page" that could be misconstrued as pages about the planet if only titles were available (as is the case with simple Web searches that provide no categorization mechanism).

Seeing that there are clear distinctions in the usage of the word Saturn, our student decides to save this organization as a profile to help filter future query results (and possibly also later look up video games as well). At this point, the student issues a new, but related, query "life on Saturn", requesting that 100 URL's be returned. These results are again passed to SONIA, but this time specifying the previously saved profile on Saturn as a categorization scheme. SONIA finds that 79 of the URL's are retrievable and classifies 9 of the documents into the category about the planet, 58 into the category on enthusiasts, and the remaining 12 in the video

Automatically Generated Descriptors	Sample Document Titles	Feasible Topics
ca, ny, street, dealers, road, avenue, nc, boulevard, mi, va, ma, pa	California Saturn Dealers New York Saturn Dealers Virginia Saturn Dealers Massachusetts Saturn Dealers	Saturn Car Dealers
saturn's, rings, ring, image, jupiter, planet, earth, plane, voyager, moons, 1995, satellites	Recent Discoveries About Saturn Voyager Images of Saturn Saturn Ring Plane Crossings of 1995-1996 Hourly Cycle of Solar System Objects	Planet Saturn
car, 1998, home, web, kind, site, cars, talk, company, 1997, copyright, cd	Saturnalia - The Saturn Enthusiasts Site Saturn And The RV Owner Saturn San Diego - Car Club Events Calendar Saturn Talk	Saturn Car Enthusiasts and Chat Groups
sega, game, games, system, >, 1998, news, video, force, #, order, quantity	SEGA SATURN – A UGO Video Game Yellow Page Video Games GamEscapes Video Games! Sega Force, Sega Saturn, Genesis, Sega CD, ... VideoGameSpot: Review Index	Sega Saturn Video Game

Table 5: Sample results on the query "Saturn".

game category. While these results may appear surprising at first, a detailed analysis of the assigned documents reveals that the classification is in fact working quite well. All 9 of the documents placed in the category about the planet are in fact about the planet. Thus, if the student were to focus on the documents that were placed in this category, he or she would be looking at only relevant pages.

On the other hand, of the 58 documents placed in the enthusiasts category we find that only 5 are really about the planet (and thus really misclassified). Most of the documents assigned to this category are actually discussions about astrology and how it affects one's "life" (hence the match to the student's query). While they are not directly about car enthusiasts, they are very much like other documents that are informally "chatting" about a subject. Finally, in the video game category, we find that of the 12 documents placed there, only 4 are really about the planet Saturn. Thus, the classification scheme, while admittedly making some misclassifications, is able to filter the vast majority of documents that are not related to the planet Saturn and consequently allows the student to focus on only those pages which are truly relevant.

While it may be argued that the student would not look at a few articles about the planet if he or she only focused on the results of a single category in the example above, this point becomes moot when we recognize the vast quantity of relevant documents that a user would never see on a subject because they are not in digital format, have not been indexed, etc. In the context of large information repositories, such as Digital Libraries and the WWW, the ability to get query results with high *precision* is generally much more important that being able to *recall* all possibly relevant documents.

Interaction model

One important aspect of the information access process that we have heretofore not discussed is the user's interaction with an interface that accesses SONIA. The SenseMaker interface (currently used with SONIA) provides a mechanism whereby users can limit a collection of documents to those categories that are of interest and then request a re-clustering of only those documents. In this way, the user can explore the information space at a variety of granularity levels and thereby quickly focus on just those few documents that are truly relevant to their information need. Note that this interaction model is very related to that of the Scatter/Gather system [6].

More significant, however, is the fact that the user can save multiple profiles during their interactions with the system and thus maintain classification schemes at several different levels of granularity. This allows the system to further bridge the gap between simple search-based systems which provide no organization for retrieved documents and hierarchical taxonomic systems which are not customized to users information needs. It is this critical issue that has prompted our work on future extensions of SONIA described below, and hence we do not give detailed examples of this interaction presently.

CONCLUSIONS

We have presented SONIA, a service that provides the ability to organize document collections either into an existing or a novel categorization scheme, using a variety of machine learning techniques. SONIA is currently integrated into the Stanford Digital Libraries Project testbed and accessible through the SenseMaker interface via the InfoBus. We have shown that SONIA can effectively help users find and keep track of relevant information in large information spaces by utilizing its automated organizational capabilities.

We are currently extending SONIA in a number of ways.

Foremost, we have constructed a new interface to the system that allows for document collections to be automatically organized into topical hierarchies rather than into a simple one level categorization structure. In this way, we hope to allow users to integrate multiple related profiles into a unified organization scheme. Moreover, by taking advantage of a hierarchical structure, we can leverage users' familiarity with existing hierarchical topical organization schemes used on the World Wide Web (e.g., *Yahoo!*) to allow users to quickly construct their own personalized and extensible hierarchies of categories.

Finally, the new system will allow even further user interactivity by allowing the user to directly manipulate the structure of the hierarchy and placement of documents within it. In this way, the system can not only help users develop new organizational schemes, but it can also help them maintain existing ones, such as their Web bookmarks.

ACKNOWLEDGMENTS

The authors thank Daphne Koller and Moises Goldszmidt for insightful discussions on this work. Additional thanks go to Marti Hearst and the anonymous reviewers for providing valuable comments on this paper. We are indebted to Scott Hassan who implemented the web crawler module of SONIA. This work was supported by ARPA/NASA/NSF under a grant to the Stanford Digital Libraries Project.

REFERENCES

1. James Allan, Anton V. Leouski, and Russell C. Swan. Interactive cluster visualization for information retrieval. Technical Report IR-116, U. Mass, Amherst, Center for Intelligent Information Retrieval, 1997.

2. Robert B. Allen, Pascal Obry, and Michael Littman. An interface for navigating clustered document sets returned by queries. In *Proceedings of ACM SIGOIS*, pages 166–171, 1993.

3. Michelle Baldonado, Chen-Chuan K. Chang, Luis Gravano, and Andreas Paepcke. The Stanford digital library metadata architecture. *International Journal of Digital Libraries*, 1(2), 1997.

4. Michelle Q. Wang Baldonado and Terry Winograd. SenseMaker: An information-exploration interface supporting the contextual evolution of a user's interests. In *Proceedings of CHI*, 1997.

5. T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

6. D. R. Cutting, D. R. Karger, J. O. Pederson, and J. W. Tukey. Scatter/Gather: a cluster-based approach to browsing large document collections. In *Proceedings of ACM/SIGIR*, pages 318–329, 1992.

7. Moises Goldszmidt and Mehran Sahami. A probabilistic approach to full-text document clustering. Technical Report ITAD-433-MS-98-044, SRI International, 1998.

8. I. J. Good. *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*. MIT Press, 1965.

9. Stanford Digital Libraries Group. The Stanford digital libraries project. *Comm. of the ACM*, April 1995.

10. Marti A. Hearst, David R. Karger, and Jan O. Pederson. Scatter/Gather as a tool for the navigation of retrieval results. In *Proceedings of AAAI Fall Symposium on Knowledge Navigation*, 1995.

11. Marti A. Hearst and Jan O. Pederson. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of ACM/SIGIR*, 1996.

12. T. Kohonen. *Self-Organizing Maps*. Springer, 1995.

13. Daphne Koller and Mehran Sahami. Toward optimal feature selection. In *Proceedings of Machine Learning*, pages 284–292, 1996.

14. Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In *Proceedings of Machine Learning*, pages 170–178, 1997.

15. P. R. Krishnaiah and L. N. Kanal. *Classification, Pattern Recognition, and Reduction in Dimensionality*. Amsterdam: North Holland, 1982.

16. David D. Lewis and M. Ringuette. Comparison of two learning algorithms for text categorization. In *Proceedings of SDAIR*, 1994.

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

18. E. Rasmussen. Clustering algorithms. In William B. Frakes and Ricardo Baeza-Yates, editors, *Information Retrieval: Data Structures and Algorithms*, pages 419–442. Prentice Hall, 1992.

19. Mehran Sahami. Learning limited dependence Bayesian classiers. In *Proceedings of KDD*, pages 335–338, 1996.

20. Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical Report 87-881, Cornell Computer Science Dept., 1987.

21. Honkela Timo, Kaski Samuel, Lagus Krista, and Kohonen Teuvo. WEBSOM - self-organizing maps of document collections. In *Proceedings of WSOM'97 Workshop on Self-Organizing Maps*, pages 310–315, 1997.

22. C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

23. Sholom M. Weiss and Casimir A. Kulikowski. *Computer Systems that Learn*. Morgan Kaufmann, 1991.