SONIA: A Service for Organizing Networked Information Autonomously

Mehran Sahami[1] **Salim Yusufali**[1] **Michelle Q. W. Baldonado**[2]

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

[2]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 Liraries 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 incomprehensibily 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 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 by 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 take the results of queries to networked information sources and, in real-time, automatically retrieve, parse and organize these documents into coherent categories for presention to the user. Moreover, the system can then save such document oragnizations 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 catetgorization scheme. In this way, it allows navigate the results of a query at a more topical level 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*) which 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 the 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, clus-

tering, 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 [5], 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 in both in navigating query results [9] as well as concentrating documents particularly relevant to a query in just one or two clusters [10]. Our system builds on this work in a number of ways.

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. Futhermore, since our system exists as part of a general architecture within the Stanford Digital Libraries Testbed [8], it has the ability to simultaneously retrieve information from a number of heterogenous sources, thereby making our system maximally flexible.

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 the 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 that are currently used in SONIA. We then show examples of the system in use, discussing its effectiveness for information browsing and classification. Finally, we give a summary of this work and its future directions.

System Overview

To get a complete picture of the 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. We then follow this with a detailed description of the components that comprise SONIA.

InfoBus Architecture

The focus of the Stanford Digital Libraries project is on providing interoperability amoung heterogeneous, distributed information sources, services and interfaces. To this end, the InfoBus architecture [1] 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 between 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 subcollections of documents that statisfy 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 (refered to *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 [2], 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 URLs (for Web documents), and the like, or it can utilize the SONIA organizational service 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, the documents will 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 their diverse information and categorization needs. To better understand the technologies incorporated within the SONIA system, we presently turn our attention to the components that comprise SONIA.

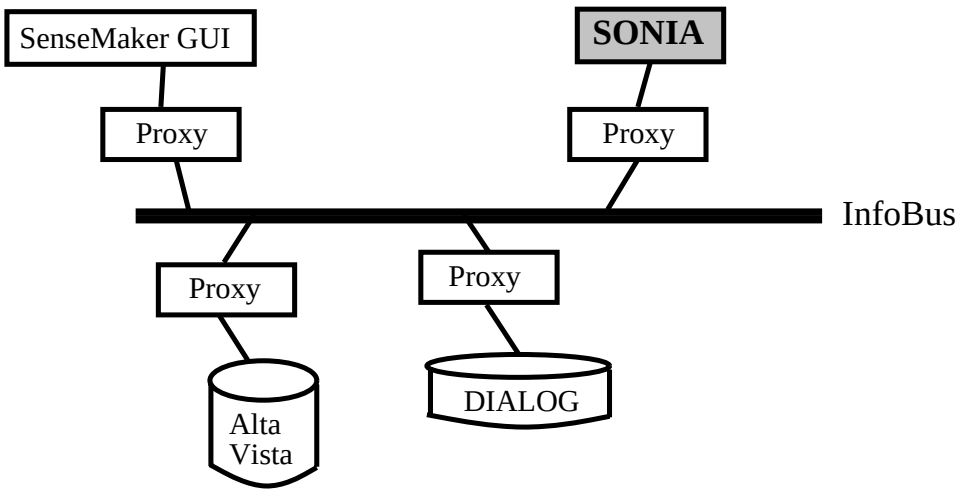

Figure 1: The InfoBus architecture.

SONIA

It is simplest to view SONIA as a chain of modules, each of which is reponsible 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 this 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 [16]. 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 [19] 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 (10^5 for even small collections), 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 currently incorporates four forms of feature selection, each of which operates on the vector-space representation of the documents. Initially, dimensions representing *stopwords* (non-meaningful terms, such as "a" and "the") are eliminated from the document vectors. These stopwords are determined using a standard English stopword list of 570 words as well as a special hand-crafted list of approximately 100 Web stopwords (such as "html" and "url").

In the second-tier of feature selection, a Zipf's Law analysis [20] of term occurrence over the collection is used. This essentially eliminates any terms that appear fewer than 3 or greater than 1000 times in the entire collection as not having adequate resolving power to differentiate subcollections of documents.

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

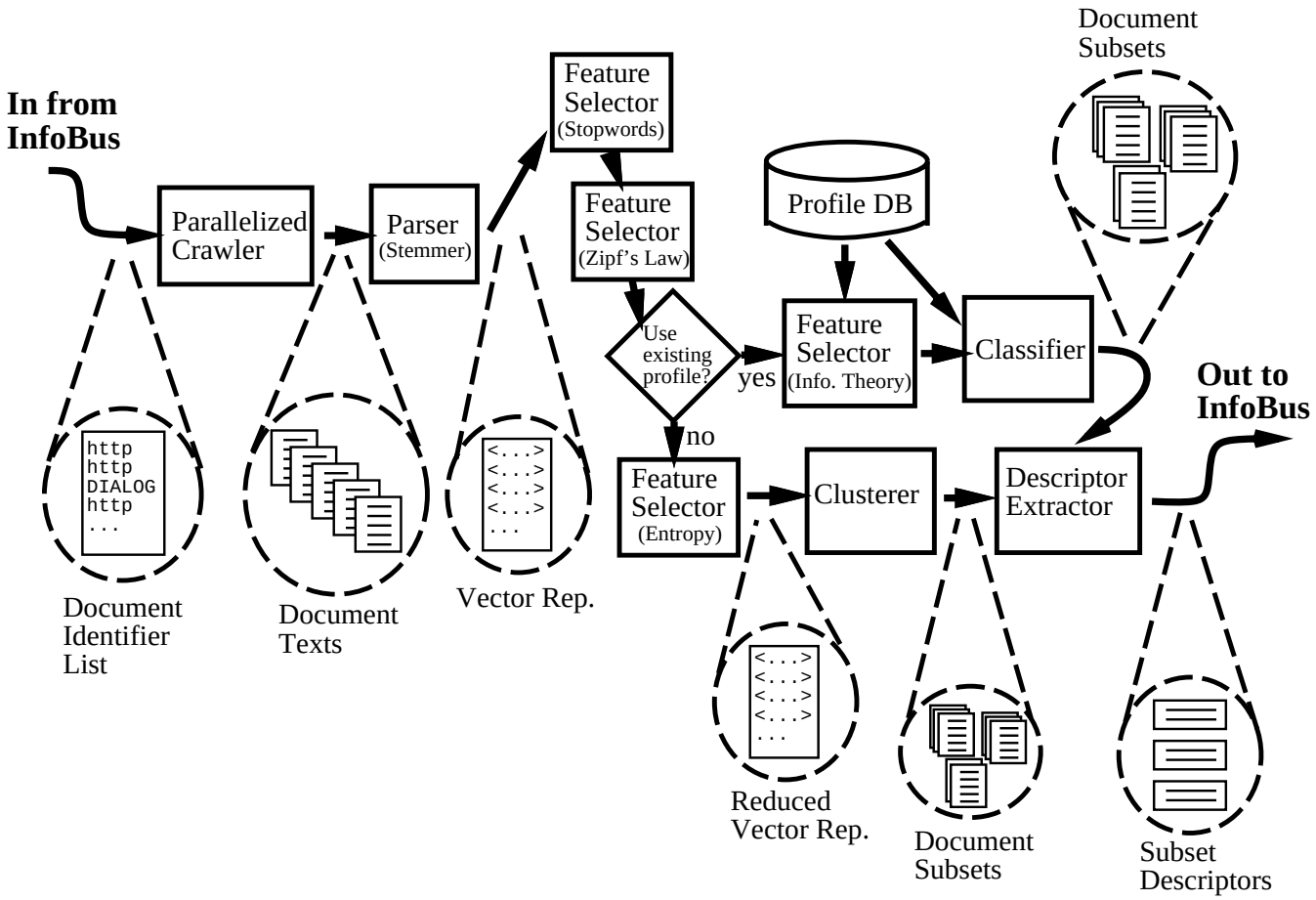

Figure 2: Processing stages in the SONIA system.

being employed. Here we have a previous organization of documents into distinct groups that has been stored in a *Profile Database* maintained by SONIA. Since we wish to organize the current set of documents according to this previous classification, 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 we have previously shown to be effective on a number of similar classification problems, including text categorization [11]. Moreover, we have also found that very few terms are needed for accurate document classification [12]. Thus, we aggresivelly 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 [4]. 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 which contains 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 an 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 hinderance.

All the feature selection methods we apply have the goal of significantly reducing the resulting feature space while focusing on those words that appear to be meaningful and have the greatest resolving power between documents.

Classification If the user chooses to categorizing documents with respect to an existing profile, we can simply cast this problem as one of classfication in the traditional machine learning sense. 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 scheme [7]. This algorithm attempts to predict for each document, d, the category, c_j, for which it has maximal probability. Formally, this is given by

$$Argmax_{c_j \in C} \frac{P(d|C = c_j) \cdot P(C = c_j)}{P(d)}$$

where C denotes the set of all possible categories. Since the value of $P(d)$ is the same regardless of catagory, 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, $t_1, t_2, \ldots, t_n$, thus making it intractable to compute $P(C|t_1, t_2, \ldots, t_n)$ directly. Rather, the Naive Bayesian classifier makes the simplifying assumption that

$$P(t_1, t_2, \ldots, t_n|C) = \prod_{i=1}^{n} P(t_i|C).$$

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 [14]. Nevertheless, to relax this restrictive assumption we have recently implemented more expressive Bayesian classification schemes [18] in SONIA, and found them to be significantly better for document classification [12].

In addition to considering other classification algorithms, we also considered using different document representations for classification as well (such as using a term-weighting approach rather than Boolean vectors). Previous research gives evidence that there may not be significant differences in classification results between these representations [21].

Regardless, 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 AutoClass [3], hierarchical agglomerative clustering [17] and iterative clustering methods, such as K-Means [13]. A complete account of these experiments is beyond the scope of this paper, but we refer the interested reader to [6].

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 similarity score between pairs of documents which, for generality, we will refer to as $Sim(d, d')$. One commonly used similarity score is simply to compute the cosine of the angle between two normalized document vectors. This is the score we use in the examples reported here.

The hierarchical method creates a clustering 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,

$$Sim(c, c') = \sum_{d \in c, d' \in c'} \frac{1}{|c| \cdot |c'|} Sim(d, d').$$

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 minium 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 iteration is performed (we used a maximum of 5 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 hardcoded value, generally between 2 and 10. Currently, we are exploring an extended interface to this system which easily allows users to vary this parameter.

Descriptor extraction Once classification or clustering have 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 of a term t_i appearing in a document in c_j versus appearing in a document in any other group:

$$O_j(t_i) = \frac{P(t_i|c_j)}{\sum_{c_k \neq c_j} P(t_i|c_k)}$$

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 stopword 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 is 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 Usage

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 a objective measure by which to measure such a system as a whole, we augment the examples with details from controlled studies in which the efficacy of the clustering and classification methods implemented in SONIA could be measured directly.

Scenario One

In the first scenario, we consider the situation in which a resercher 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 URLs are returned. These URLs are then passed (again through the InfoBus) to SONIA without specifying an exising profile for classification. The crawler in SONIA is able to retrieve 141 valid Web pages from these URLs in the allotted lookup time. These pages are then immediately 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 catagorizing these documents, they are clustered to form a new organizational scheme. The result of this clustering (into 4 catagories) is shown in Table 1, 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 that a user may spend almost that long waiting for just a few pages to load if they were browsing manually.

From the results in Table 1, we can see that SONIA is

Automatically Generated Descriptors	Sample Document Titles	Feasible Topics
information, stanford, digital, ketchpel, http, library, user, work, steven, infobus, university, //www	Quarterly Report Stanford Digital Library Project Agent Projects in the Stanford Digital Library Home Page - Steven Ketchpel STARTS	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, university, area, design, faculty, david, science, systems, engineering), (electrical, stanford, professors	CSL 1998 EE Quals Computer Science *(departmental page)* Faculty of the Center for Telecommunications Journal of the ACM Editorial Board	Professorial 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 1: Sample results on the query "Hector Garcia-Molina".

effective at picking out that 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 colleages 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 help the user quickly eliminate irrelevant documents that just happen to match their query. These results are consistent with previous findings that show this clustering technology is quite effective at recovering a known structure in a document collection [6]. We have found error rates for such structure recovery to range from 1% to 30% in terms of misclassified documents. Moreover, others have shown that clustering can effectively convey the structure of a document collection to users [15].

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 to find out what general area Sudarshan is working in and thus only requests the top 30 URLs from the search service. The user then requests that SONIA classify these resulting URLs according to the previously saved "Hector" profile. Here we find that SONIA is able to retrieve 29 valid documents and classifies all of them in the category pertaining to Database Research. A subsequent analysis of the actual documents reveals that Sudarshan does in fact work on distributed database systems and is not actually a member of the Stanford Digital Libraries Project. Moreover, of the 29 documents, 25 refer specifically to research, conferences and colleagues in the area of database systems. The remaining four documents are index pages for graduate students at Stanford (presumably several of which are working on database systems) and one on housing options at Stanford. While it is arguable that these pages might have been classified in the "Professorial Duties" category, it is unclear at best. 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 topically organized automatically.

In controlled settings, we have also found that the combination of feature selection and automated document classification can be quite successful for filtering new information into the proper catagory [12]. We have observed classification accuracies to range from approximately 80% up as high as 95% in applying the same technology used in SONIA to the classification of Reuters

newswire articles.

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 URLs. As before, these URLs 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 URLs, 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 2.

As before, 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 easily be misconstrued as a page about the planet if only titles were available, as is the case with simple Web searches which 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 query "life on Saturn", requesting that 100 URLs 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 URLs 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 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 catagory 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, they 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 discusssions about astrology and how it effects ones "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, was able to filter the vast majority of documents that were not related to the planet Saturn and thus allow 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 they 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 on the Web and in Digital Libraries, 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 [5].

More significantly, 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 topical index 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 vari-

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 Virgina Saturn Dealers Massachusetts Saturn Dealers	Saturn Car Dealers
saturn's, rings, ring, jupiter, image, planet, earth, plane, moons, voyager, 1995, moon	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, automatic	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 2: Sample results on the query "Saturn".

ety of machine learning techniques. SONIA is currently integrated into the Stanford Digital Libraries Project testbed and accesible 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 a constructing a new interface to the system that allows for document collections to be automatically organized into topical hierarchies rather than a simple one level categorization structure. In this way, we hope to allow users to integrate multiple related profiles which they may construct 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 (i.e., *Yahoo!*) to allow users to quickly construct their own personalized and extensible hierarchy 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 documents placement 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.

Acknowledgements

The authors thank Daphne Koller for insightful discussions on this work and Marti Hearst for providing valuable comments on a previous version of 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. 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.

2. 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.

3. P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman. AutoClass: a bayesian classification system. In *Proceedings of Machine Learning*, pages 54–64, 1988.

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

5. 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.

6. Moises Goldszmidt and Mehran Sahami. A probabilistic approach to full-text document clustering. In preparation, 1998.

7. Irving John Good. *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*. M.I.T. Press, 1965.

8. Stanford Digital Libraries Group. The stanford digital libraries project. *Communications of the ACM*, April 1995.

9. 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 Sympposium on Knowledge Navigation*, 1995.

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

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

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

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

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

15. Peter Pirolli, Patricia Schank, Marti Hearst, and Christine Diehl. Scatter/gather browsing communicates the topic structure of a very large text collection. In *Proceedings of CHI*, 1996.

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

17. 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.

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

19. Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical Report 87-881, Cornell University Computer Science Department, November 1987.

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

21. Yiming Yang and Christopher G. Chute. An example-based mapping method for text categorization and retrieval. *Transactions of Office Information Systems*, 12(3), 1994. Special Issue on Text Categorization.