Discovering Recurring Anomalies in Text Reports Regarding Complex Space Systems

Ashok N. Srivastava, Ph.D. and Brett Zane-Ulman

Abstract—Many existing complex space systems have a significant amount of historical maintenance and problem data bases that are stored in unstructured text forms. For some platforms, these reports may be encoded as scanned images rather than even searchable text. The problem that we address in this paper is the discovery of recurring anomalies and relationships between different problem reports that may indicate larger systemic problems. We will illustrate our techniques on data from discrepancy reports regarding software anomalies in the Space Shuttle. These free text reports are written by a number of different people, thus the emphasis and wording varies considerably.

We test three automatic methods of anomaly detection in text which are popular in the current literature on text mining. The first method that we describe is k-means or Gaussian mixture model and its application to the term-document matrix. The third method is based on an analysis of the results of applying a new clustering method, based on the von Mises Fisher distribution, that represents each document as a point on a high dimensional sphere. In this space, we perform clustering to obtain a set of potential anomalies. We also describe results that are derived from a new method known as spectral clustering, where vectors from the term-document matrix are embedded in a high dimensional space for clustering.

The paper concludes with recommendations regarding the development of an operational text mining system for analysis of problem reports that arise from complex space systems. We also contrast such systems with general purpose text mining systems, illustrating the areas in which this system needs to be specified for the space domain.

TABLE OF CONTENTS

This work was supported by the NASA Intelligent Systems Intelligent Data Understanding Program. A. N. Srivastava is at the NASA Ames Research Center. B. Zane-Ulman is with the Computer Sciences Corporation at NASA Ames.

1. INTRODUCTION

Many complex aerospace systems have a variety of prognostic and diagnostic instrumentation that deliver high speed data streams of information regarding the current health of the system. These streams give instantaneous information about the system and must be analyzed accordingly.

Along with these data streams, however, aerospace systems also have significant maintenance records associated with them. These maintenance records are often free text reports. They are often recorded by maintenance personnel or engineers that are responsible for specific subsystems in the vehicle. In some cases, such as the Aviation Safety Reporting System [1], the reports are augmented by some structured data through the use of a coded report. The coded reports can be analyzed using standard statistical methods or data mining methods that are suited for the analysis of structured information.

The free text reports, however, need to be significantly transformed to be analyzed with standard data mining or statistical methods. Most of those methods assume that the data can be expressed as a matrix where each row is an observation and each column is a variable. For example, in the case of analyzing the variations in reliability for 1000 different thermal sensors, a matrix could be formed which would have 1000 rows and columns corresponding to various reliability metrics as well as other information regarding the sensors that are deemed relevant by the analyst. This information could include, for example, where the sensor was manufactured, when it was manufactured, information regarding the manufacturing process, etc. These pieces of information would form the columns of the data matrix that could then be submitted to a statistical or data mining analysis.

This paper discusses methods of analyzing free text documents where the text is represented in a matrix as described above– each document corresponds to a row in the matrix, and the columns correspond to the union of all the key words in all the documents. The entries in the matrix (called a term-document matrix) correspond to the frequencies of each key word (or term) in the document. Through this procedure each document is represented in a point in a high dimensional vector space. This representation is used by many text analysis methods under the terms 'bag-of-words', latent semantic analysis, and other research areas [2]. A significant drawback of this vector space approach is that all semantic and syntactic information in the document is lost.

In the next section, we motivate the particular problem that

we use to demonstrate our methodology, which is detecting recurring anomalies in free text reports that are used in the Space Shuttle's Flight Readiness Reviews. In Section III, we describe two methodologies that are standard for analyzing text documents that are represented in vector spaces. In Section IV, we briefly review the relevant algorithms. In Section V we discuss our experimental results and in Section VI we conclude the paper by summarizing it and discussing future work.

2. Discovering Recurring Anomalies

The problem that we address in this paper is as follows. Given a set of N documents, where each document is a free text English document that describes a problem, an observation, a treatment, a study, or some other aspect of the the vehicle, automatically identify a set for potential recurring anomalies in the reports. Note that for many applications, $N \approx 100,000$, which is a corpus that is too large for a single person to read, understand, and analyze by hand. Thus, while engineers and technicians can and do read and analyze all documents that are relevant to their specific subsystem, it is possible that other documents, which are not directly related to their subsystem still discuss problems in the subsystem. While these issues could be addressed to some degree with the addition of structured data, it is unlikely that all such relationships would be captured in the structured data. Therefore, we need to develop methods to uncover recurring anomalies that may be buried in these large text stores.

One approach to addressing this problem would be to develop a method to query the text database for known anomalies. For example, one could envision generating a list of queries, such as "find all examples of software errors", or "find all examples of navigation system faults", etc. While such a query mechanism is useful, it still does not address the problem of finding anomalies that may not be thought of a priori. The approaches that we describe in this paper are particularly useful for identifying unknown recurring anomalies.

The methods that we use to discover these anomalies are based on various clustering methods. *Clustering* refers to the process of identifying subsets of rows in the term-document matrix that have similar characteristics. The first approach that we discuss is based on the k-means clustering algorithm of the term-document matrix which implicitly makes Gaussian assumptions and uses the Euclidean distance between term-document vectors as a measure of similarity. The second clustering method uses the cosine measurement between two vectors and which implicitly assumes the von Mises Fisher distribution. The third clustering method, based on spectral clustering, embeds the term-document vectors in an infinite dimensional space and looks at the clustering of a low dimensional projection. These formulations will be discussed in the next section.

Our procedure to identify recurring anomalies is based on the idea that they will show up in the same cluster, and thus is highly dependent on the clustering algorithm. In this section, we describe three methods of cluster analysis that are popular in the literature today and discuss their underlying assumptions. These assumptions affect the outcome of the clustering and therefore can affect the discovery of recurring anomalies.

For purposes of the discussion presented here, we will model the text as a term-document matrix [3]. The term-document is described by an $N \times p$ matrix Z, where N is the number of documents, and p is the number of keywords in the union of all documents. A keyword is defined as a word that is informative about the content of the document. Words such as 'and', 'the', 'but', and 'not' are called stop words and are abandoned when the term-document matrix is created. In many applications, $p \gg N$. In order to remove terms from this matrix that have small frequencies as compared to the number of documents, it is customary to perform a data reduction technique known as *Term Frequency Inverse Document Frequency* (TFIDF) to the term document matrix. We follow the notation in [3] as follows. For Z_ij, which corresponds to the entry in the matrix for the ith document d_i and the jth term t_j, TFIDF is a straightforward procedure and can be computed as follows:

$$Z_{ij} = TF(t_j, d_i) \times IDF(t_j) \tag{1}$$

$TF(t_j, d_i)$ is the term frequency, which is the frequency that term t_j appears in document d_i. $IDF(t_j)$ is the Inverse Document Frequency of term t_j and is defined as:

$$IDF(t_j) = \log(\frac{N}{DF(t_j)}) \tag{2}$$

where $DF(t_j)$ is the number of times that term t_j appears in the corpus. Notice that if this number is close to N, the number of documents in the corpus, $IDF(t_j) \approx 0$, and the term's contribution to the matrix is very small.

Dimensionality Reduction using Principal Components Analysis

Insert text here.

k-means Algorithm and Mixture Models

The k-means clustering algorithm [4] is perhaps the most popular method of clustering structured data due to its simplicity of implementation. The algorithm works by choosing k random initial cluster centers, computing the distances between these cluster centers and each row in the data matrix and then identifying those rows that closest to each cluster center. The corresponding cluster centers are moved to the centroid of those data points and the procedure is repeated. The algorithm converges when the cluster centers do not move from one iteration to the next.

The k-means algorithm is a special implementation of the Gaussian Mixture Model. These models assume that the data vectors are generated according to the probability den-

sity $P(Z_i|\Theta)$ and assume that:

$$P(Z_i|\Theta) = \sum_{c=1}^{C} P(c)P(Z_i|\theta_c) \quad (3)$$

where Θ is a vector containing the C model parameters, and θ_c are the model parameters for the cth mixture component. The vector Z_i is a p dimensional vector from the term-document matrix. The parameters of this model are obtained through Expectation Maximization of the appropriate log-likelihood function or, more generally, the posterior log-likelihood. In the case of a Gaussian mixture density model for $Z_i \in \mathcal{R}^l$, we take the likelihood function as:

$$\begin{aligned} P(Z_i|\theta_c) &= P(Z_i|\mu_c, \Sigma_c, c) \\ &= (2\pi)^{-\frac{d}{2}}|\Sigma_c|^{-\frac{1}{2}} \times \\ &\quad \exp[-\frac{1}{2}(Z_i - \mu_c)^T \Sigma_c^{-1}(Z_i - \mu_c)] \end{aligned}$$

Maximum a posteriori estimation is performed by taking the log of the posterior likelihood of each data point Z_i given the model Θ using the Expectation Maximization algorithm [5].

In the case of text clustering the vectors are high dimensional and sparse. In this case, the k-means algorithm does not work well because the number of data points needed to from dense regions increases exponentially with the number of dimensions. With a finite amount of data and high dimension, most data points end up being approximately equidistant to each other. Figure 1 shows the effect of increasing the number of dimensions on the average Euclidean distance between points. The x-axis is a logarithmic scale from 10 to 10,000 dimensions. From top to bottom, the curves indicate the effect of sparseness of the vectors. The bottom curve corresponds to a vector that is 50% sparse, the remaining curves correspond to vectors that are 66%, 75%, and 90% sparse, respectively.

Sammon Nonlinear Mappings

Insert text here.

von Mises Fisher Clustering

The Gaussian Mixture Model and k-means algorithms make Gaussian assumptions about the underlying distribution of the data. Empirical studies have shown that for high dimensional sparse data sets, the cosine measure of similarity between two vectors is a better measure than the Euclidean distance. A recent paper [6] developed the mathematics to perform clustering using the cosine measure of similarity. Just as the Euclidean distance implicitly implies a Gaussian distribution the cosine distance implicitly implies a different distribution, known as the von Mises Fisher distribution. We follow the formulation in [6] closely:

$$P(Z_i|\Theta) = \sum_{c=1}^{C} P(c)P(Z_i|\theta_c) \quad (4)$$

In this case, we assume that the vectors Z_i have been normalized to unit length. For p dimensional data vectors, we have

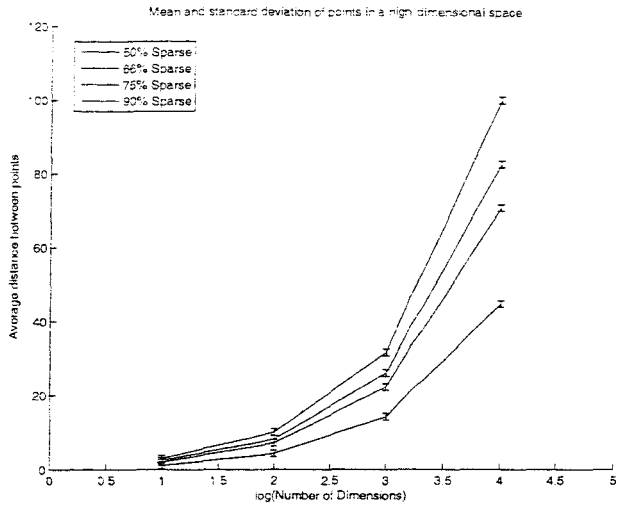


Figure 1. The effect of sparseness and increasing number of dimensions on the Euclidean distance between points. Notice that for a given sparseness (corresponding to one curve) the average distance between points increases with the number of dimensions.

the von Mises Fisher (vMF) distribution:

$$P(Z_i|\mu, \kappa) = c_p(\kappa)\exp(\kappa\mu^T Z_i) \quad (5)$$

where μ is a unit vector corresponding to the mean of the distribution and $\kappa \geq 0$ is the measure of dispersion. The constant $c_p(\kappa)$ is given by:

$$c_p(\kappa) = \frac{\kappa^{(p/2)-1}}{(2\pi)^{(p/2)} I_{(d/2-1)}(\kappa)} \quad (6)$$

where $I_r)(\kappa)$ represents the modified Bessel function of the first kind of order r. With the vMF distribution as defined above, Banerjee et. al. 2003 derive the Expectation Maximization algorithm to optimize a mixture of vMF distributions. Their results indicate that this algorithm has superior performance on high dimensional text clustering problems compared to the k-means algorithm.

Spectral Clustering

Spectral clustering is a different approach to clustering that works by embedding the vectors Z_i in a high, possibly infinite dimensional space using Mercer Kernels [7]. Mercer Kernel functions can be viewed as a measure of the similarity. For a finite sample of data $\mathcal{Z}$, the kernel function yields a symmetric $N \times N$ positive definite matrix, where the (i, j) entry corresponds to the similarity between (Z_i, Z_j) as measured by the kernel function. Because of the positive definite property, such a Mercer Kernel can be written as the inner product of the data in the feature space. Thus, if $\Phi(Z_i) : \mathcal{R}^l \mapsto \mathcal{F}$ is the (perhaps implicitly) defined embedding function, we have $K(Zi, Z_j) = \Phi(Z_i)\Phi^T(Z_j)$. Typical kernel functions include the Gaussian kernel for which $K(Z_i, Z_j) =$

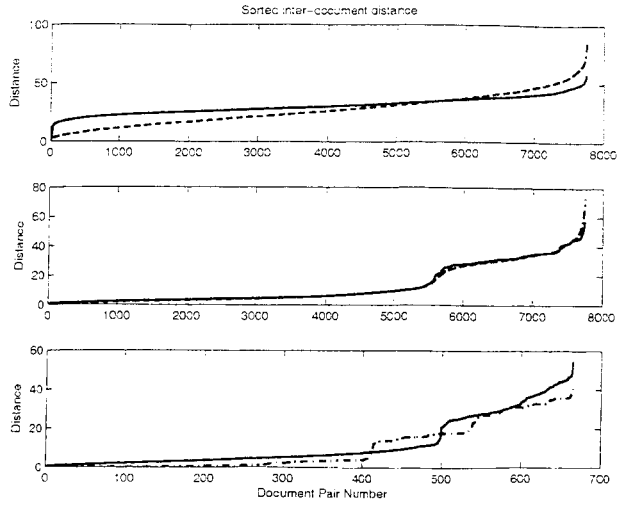


Figure 2. The top panel of this plot shows the sorted inter-document distances in the original 500 dimensional space and the distances that arise from a 2 dimensional approximation to the original distances. Original distances are shown in the solid line, and the dotted line shows the distances with the 2 dimensional approximation. Notice that there is substantial error in the approximation. The middle panel shows the results of Sammon mapping after the dimension of the document space is reduced from 500 dimensions to 10 dimensions using principal components analysis. The agreement between the distances in the low dimensional space and the 2 dimensional mapping are excellent. The bottom panel shows the approximation of the Sammon mapping using a neural network.

$\Phi(Z_i)\Phi^T(Z_j) = \exp(-\frac{1}{2\sigma^2}||Z_i - Z_j||^2)$, and the polynomial kernel $K(Z_i, Z_j) = \Phi(Z_i)\Phi^T(Z_j) =< Z_i, Z_j >^p$.

For supervised learning tasks, linear algorithms are used to define relationships between the target variable and the embedded features [8]. Work has also been done in using kernel methods for unsupervised learning tasks, such as kernel clustering [9], [?] and density estimation [10].

Spectral clustering works by computing the eigenvectors of a normalized kernel matrix (see [7] for details of the algorithm). The largest n eigenvectors are chosen and normalized to unit length. The rows of the eigenvectors (corresponding to N points in an n dimensional space) are then clustered using the k-means algorithm.

In Table 1, we show the asymptotic efficiency of the algorithm.

3. Developing an Operational Text Mining System

In this section we describe a system architecture for an operational text mining system. An aerospace vehicle is a highly complex system with complex interactions between its various subsystems. To get the most out of a problem tracking system as many of these complex relationships as possible need to be included in the tracking and analysis of issues that arise. The system we propose contains an engineering model of the vehicle detailing the relationship between vehicle components and subsystems. This model is joined to a relational database containing additional vehicle component information as well as structured fields for entering problem reports. This information gives the system a better context in which to do clustering of the problem reports, thereby increasing the likelihood that meaningful clusters will be produced. The need for this type of organized, interconnected structure was found to be important in the NASA Space Shuttle program by

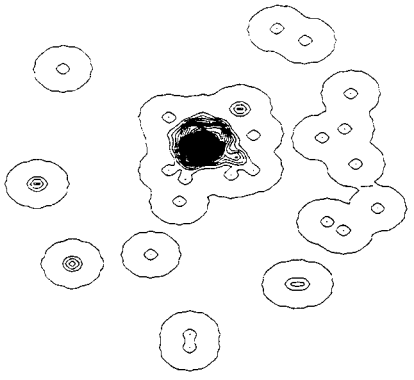


Figure 3. This visualization is a projection of the 125 dimensional document vectors into two dimensions using Sammon mapping. The contours represent regions of equiprobability. Recurring anomalies can be documents that fall within the same closed contour.

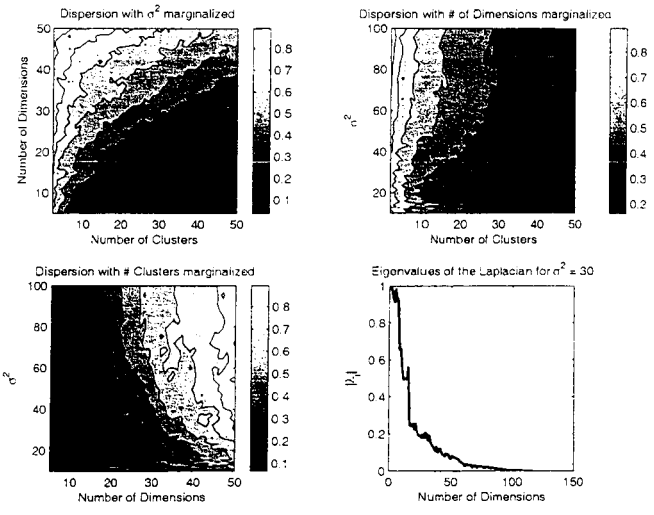


Figure 4. This visualization shows how the clustering results vary with the three parameters in spectral clustering: the number of dimensions, the number of clusters, and σ^2, which is the scale parameter in the kernel.

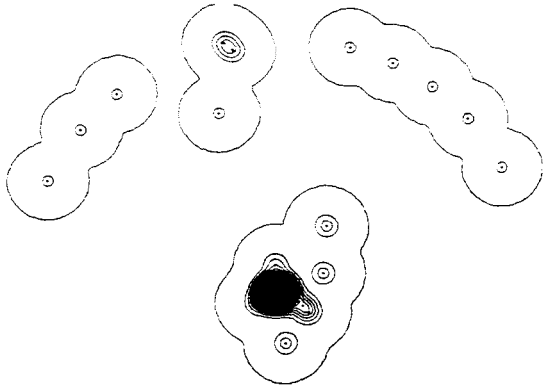


Figure 5. This visualization is a projection of the 500 dimensional document vectors into two dimensions using Sammon mapping. The contours represent regions of equiprobability. Recurring anomalies can be documents that fall within the same closed contour.

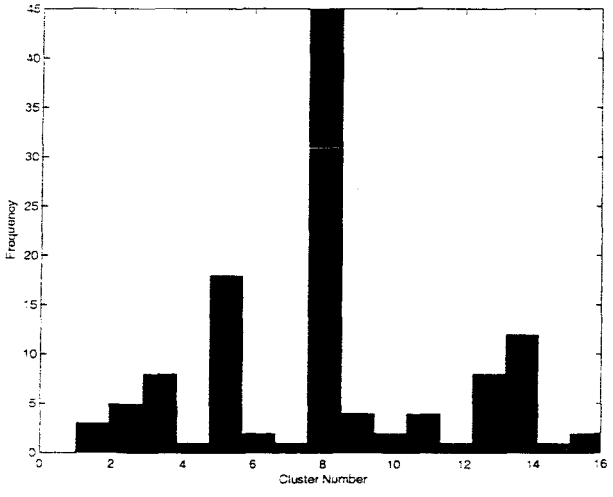


Figure 6. This diagram shows the frequency distributions of documents with clusters. The large cluster corresponds to the dense region in the previous figure.

an independent assessment team [11].

The vehicle model should consist of ontology of the language used to describe the vehicle and its components, domain information, and vehicle system structure.

The ontology portion defines the language of terms used when describing problems with the vehicle. This includes acronym definitions, thesaurus terms, conceptual hierarchies, and irrelevant terms. Commonly used acronyms need to be defined so that their terms can be related between documents with related, but not identical, references. A set of thesaurus terms will help to relate documents by their intended meaning, not just their literal content. Conceptual hierarchies group sets of terms into low level concepts, and low level concepts into higher level concepts. It can be thought of as a tree structure, with all of the terms in the language at the leaves of the tree. The parent node of a set of terms is the concept shared by all of those terms. At the next level up the tree these low level concepts are joined by a parent node which groups them into a higher level concept. This continues up the tree (hierarchy) to the root node, which joins all the highest level concepts together and represents the base concept of the entire language. This hierarchical structure helps to put terms in context and to create links between documents. The weight of the links can be varied depending on the level in the conceptual hierarchy that the link was made. A set of irrelevant terms should also be included in the vehicle model for common terms or codes that shouldn't be used when clustering documents.

Table 1. Asymptotic Efficiency of the Detection Algorithm Vs. SNR

q	1.0	1.5	2.0	2.5	3.0	3.5
EF_1	1.73	1.31	1.01	0.79	0.64	0.53
EF_0	5.17	2.30	1.29	0.83	0.57	0.42

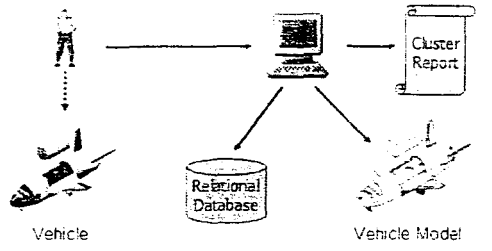


Figure 7. System Architecture - Engineers observing the vehicle enter problem reports into the system, which are stored in a relational database and joined with vehicle component and design information. This allows for flexible reporting and more relevant analysis of trends in the problem reports.

The domain information consists of relationships between terms. Terms can be related by causality (ie. 'water' causes 'corrosion'), similarity, mutual exclusivity, etc. These relationships should describe physical and engineering relationships that are specific to the vehicle design.

The vehicle system structure is an engineering model that defines how parts, components, and subsystems interact with each other.

The relational database consists of tables for all of the vehicle parts, components, and subsystems. It also has transactional tables for entering problem reports with both fixed fields and free text fields. Setting up the database for problem tracking in this manner will allow for simple and complex queries to

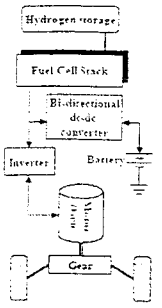


Figure 8. Example vehicle system structure - specifies how components and subsystems fit together so that analysis methods can take into account interactions between subsystems.

answer common high level questions as well as give a great deal more information to the clustering algorithms.

The part table should have a part ID as a primary key. Each record should be a unique part with fields describing properties of the part as well as component IDs of each component the part is used for in the vehicle. These can be used for joining with the component table.

Similarly, the component table should have a component ID as the primary key and each record should describe properties of the component. It should list subsystem IDs of each subsystem the component is part of.

The problem report tables are similar to bug tracking systems commonly used in software development. They should consist of fixed fields for things like title, priority, problem category, severity, and subsystem or component if applicable. The free text field is the main body of the problem report where a full description and discussion of the problem is entered.

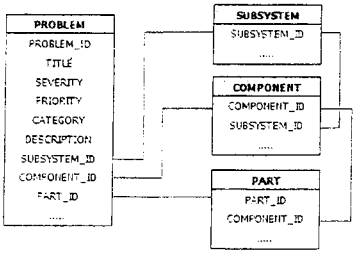


Figure 9. Relational Database Schema - joins problem reports with detailed information about the vehicle's parts, components, and subsystems.

The system can be used to perform clustering of problem reports in order to discover recurring anomalies. As observers discover problems on the vehicle they enter them into the relational database through a simple web interface. If the vehicle system design changes then the vehicle model is updated to reflect those changes. Regular reports of open issues can be generated simply by querying the database. A streamlined, efficient process flow for entering and analyzing problem reports is critical for analyzing trends and recurring anomalies [12]. To this end, the system architecture, clustering algorithms, and methods described above can be used.

4. Data Analysis

These are the steps we took to create clusters from a set of Flight Readiness Reports and Discrepancy Reports for the space shuttle. We first converted the documents into plain text format. Some documents were scanned images of printed pages. For these we performed OCR (Optical Character Recognition) to convert the images to text.

Since we had a small number of documents, and there were repeating sections throughout the documents, we broke the documents apart into sections which were treated as independent from one another.

We next created a Bag of Words matrix. This matrix has a column for each keyword appearing in the entire collection of documents and a row for each document. For each row vector, which represents a single document, the number of occurrences of each keyword is placed in the appropriate column of the matrix.

In order to weight distinctive terms properly during the clustering we then applied tfidf.

For some algorithms the dimensionality (number of columns) of the bag of words matrix was too high. We reduced the dimensionality by performing Singular Value Decomposition and selecting just the fist 30 dimensions.

Each document vector was normalized by dividing each element by the vector's L2 norm.

We were then able to apply the clustering methods described above.

5. Conclusions

For the Discrepancy Reports we also were given a set of groupings that was done by shuttle software team members. We were able to compare our own clustering results with these. We found that in several cases, documents that were very similar had been grouped in separate clusters by the software team members but were identified with the same cluster by our clustering software.

6. Acknowledgements

The authors thank Dr. X for useful discussions. The research was supported in part by the U.S. ONR grants #.

REFERENCES

[1] L. Connell, "Incident reporting: The nasa aviation safety reporting system," *GSE Today*, pp. 66–68, 1999.

[2] T. K. Landauer, D. Laham, and P. Foltz, "Learning human-like knowledge by singular value decomposition: A progress report," in *Advances in Neural Information Processing Systems*, M. I. Jordan, M. J. Kearns, and S. A. Solla, Eds., vol. 10. The MIT Press, 1998. [Online]. Available: citeseer.ist.psu.edu/landauer98learning.html

[3] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization," in *Proceedings of ICML-97, 14th International Conference on Machine Learning*, D. H. Fisher, Ed. Nashville, US: Morgan Kaufmann Publishers, San Francisco, US, 1997, pp. 143–151. [Online]. Available: citeseer.ist.psu.edu/joachims96probabilistic.html

[4] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm, Tech. Rep. AIM-1440, 1993. [Online]. Available: citeseer.ist.psu.edu/article/jordan94hierarchical.html

[5] A. P. Dempster, M. Laird, N., and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society B*, 1977.

[6] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, "Generative model-based clustering of directional data," 2003. [Online]. Available: citeseer.ist.psu.edu/article/banerjee03generative.html

[7] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," 2001. [Online]. Available: citeseer.ist.psu.edu/ng01spectral.html

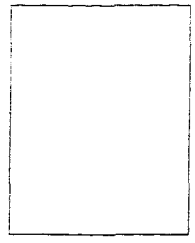
[8] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[9] M. Girolami, "Mercer kernel based clustering in feature space," *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 780–784, 2001.

[10] W. G. Macready, "Density estimation with mercer kernels," *Technical Report TR03.13 of the Research Institute of Advanced Computer Science*, 2003.

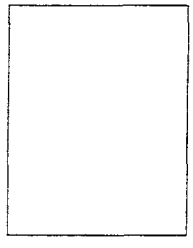
[11] H. McDonald, "Shuttle independent assessment team report," Splace Shuttle Independent Assessment Team Report to Associate Administrator Office of Space Flight, Tech. Rep., 1999. [Online]. Available: www.hq.nasa.gov/osf/siat.pdf

[12] C. Linde and R. Wales, "Work process issues in nasa's problem reporting and corrective action (praca) database," NASA Ames Research Center, Human Factors Division, Tech. Rep., 2001. [Online]. Available: human-factors.arc.nasa.gov/april01-workshop/2pg-linde3.doc



Ashok N. Srivastava *is a Principal Scientist and Group Leader in the Data Mining and Complex Adaptive Systems Group at NASA Ames Research Center. He has fourteen years of research, development, and consulting experience in machine learning, data mining, and data analysis in time series analysis, signal processing, and applied physics. Dr. Srivastava has had significant experience both in research (NASA, NIST, IBM) as well as the business world at IBM (Senior Consultant) and Blue Martini Software (Senior Director).*

Dr. Srivastava's machine learning research interests include topics in kernel methods, assessment of linear and nonlinear covariability, understanding and forecasting time-based data, and image processing. He is also interested in distributed data mining and scalability issues in federated data systems. A primary area of applied research is in the development of onboard satellite algorithms for automatic detecting and discovery of geophysical processes.



Brett Zane-Ulman *is a Computer Scientist at Computer Sciences Corporation (CSC), in the Data Mining and Complex Adaptive Systems Group at NASA Ames Research Center. He has seven years of software development and consulting experience in data mining, and data visualization. He has developed enterprise software at SGI and Blue Martini Software.*