AIAA Infotech@Aerospace 2007 Conference and Exhibit
7 - 10 May 2007, Rohnert Park, California

AIAA 2007-2834

Automatic Discovery of Anomalies Reported in Aerospace Systems Health and Safety Documents

J. P. Castle*

Mission Control Technologies, El Segundo, CA, 90245, USA

J. C. Stutz† and D. M. McIntosh†

NASA Ames Research Center, Moffett Field, CA, 94035-1000, USA

Most complex aerospace systems involve large numbers of text reports relating to safety, maintenance, and associated issues. Some have thousands of reports, spanning decades. The Space Shuttle has over 100,000 reports from just the last two decades. Similarly, the Aviation Safety Reporting System (ASRS) database spans several decades and contains over 600,000 reports.

These information repositories contain valuable information about system health, particularly about trends and recurring problems. However, repository volume and complexity can make human analysis difficult. Current methods for identifying recurring anomalies rely on a divide-and-conquer strategy. A system is decomposed into subsystems, and experts on those subsystems read and monitor current reports. Thus problems and anomalies can be tracked and trended.

This methodology requires experts who can recall and integrate reports spanning potentially long time scales. Current reports are often related to ones from months and years past. Recall is aided by categorizing reports into specific anomaly categories. Tracking and trending are aided by monitoring category totals over time. Category utility, however, relies on the experts to correctly and consistently categorize reports. With multiple experts, inter-rater reliability must somehow be assured.

Clearly, human reading, comprehension, and association with relevant prior reports, is essential to system health assurance. The weak point is in identification and recall of relevant prior reports. A decision support system that automatically analyzes reports and provides consistent discovery, characterization, and categorization would be extremely useful.

This paper discusses recent innovations in the field of text mining that enable the automatic discovery of anomalies in such text repositories, using statistical and content-based clustering techniques, and the characterization and categorization of these anomalies using advanced classification algorithms. The first innovation discovers recurring anomalies using content-based and statistical clustering techniques. The system, known as the Recurring Anomaly Detection System (ReADS) performs comprehensive analysis of an entire information repository to discover recurring anomalies and presents the results in an intuitive interactive visualization for further investigation by experts. The second innovation, known as Mariana, automatically classifies documents into predetermined categories using an advanced classifier, known as a Support Vector Machine, along with a Markov Chain Monte Carlo simulation to find the best hyperparameters for the model. Another approach to finding predetermined categories, base on Non-negative Matrix Factorization, is also showing promising results. We discuss the application of these to the problems of discovering recurring anomalies and categorizing anomalies in space and aeronautics domains.

*Computer Scientist, Intelligent Systems Division.
†Computer Engineer, Intelligent Systems Division.

Copyright © 2007 by the American Institute of Aeronautics and Astronautics, Inc.
The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental purposes.
All other rights are reserved by the copyright owner.

I. Introduction

Most complex aerospace systems have large numbers of reports relating to requirements, design, manufacturing, operation, maintenance, safety, and associated issues. Importantly, these systems also usually have a formal problem reporting system. Problem reporting systems can be quite large, containing thousands of reports, spanning many years, and could even bridge several databases. The Space Shuttle problem reporting system has accumulated over 100,000 reports from the last two decades alone. The Aviation Safety Reporting System (ASRS), despite its constrained scope, spans several decades and contains over 600,000 reports.[1] These two examples probably represent the upper end of system report library sizes; numbers around 1000 are much more common. But even in smaller datasets, the sheer number of reports tends to reduce the amount of knowledge that can easily be extracted from these reports, thereby limiting their utility.

Such information repositories contain much valuable information about many aspects of both the subject system and any similar systems. However, repository volume and complexity makes human analysis difficult. Our primary concern is system health and safety, particularly system trends and recurring anomalies for prediction, amelioration and prevention of ongoing and future system problems. This entails particular difficulties for human analysts. Current methods for identifying recurring anomalies rely on a divide and conquer strategy. A system is decomposed into subsystems, and experts of those subsystems read and monitor reports. With very small datasets, subsystem problems and anomalies can be directly tracked and trended, though subsystem interactions may still remain a problem area.

For trend and recurring anomaly detection on large datasets or across multiple datasets, this methodology requires experts who can recall and integrate information from reports spanning potentially long time scales. Current reports are often related to ones from months and years past. Recall is aided by categorizing reports into specific anomaly categories, such as "Propulsion Anomaly" or "Ground Proximity Warning System Anomaly". Tracking and trending are aided by monitoring category totals over time. Category utility relies foremost on having an informative set of categories, one that will highlight known potential problem areas without hiding unknown ones. Category utility relies almost equally on having experts who can correctly and consistently categorize reports. With multiple experts, inter-rater reliability must somehow be assured and rating "drift," the tendency for analysts to morph their ratings over time, must be carefully monitored.

While it is clear that system health assurance depends on human comprehension and association with relevant prior reports, a weak point is in identification and recall of relevant prior reports. A decision support system that automatically analyzes reports and provides consistent discovery, characterization, and categorization would be extremely useful.

The scenarios described above can be broken down into two sets of problems: those types of systems with well-defined problem categories, as found in the aviation community; and those whose nature does not yet lend itself to classification and must therefore depend on unsupervised techniques, such as those in the aerospace community. Both types will be addressed in this paper.

II. Related Work and Research

Unsupervised text mining methods, such as document clustering and nearest neighbor approaches are well suited for monitoring multiple large datasets. Several clustering techniques were explored in the early-stage of development of the Recurring Anomaly Identification System (ReADS). As explained in the Srivastava et al. paper, a modified version of von Mises Fisher clustering was ultimately selected for the unsupervised clustering component. A similar comparison study was done to determine an appropriate method for determining recurring anomalies. Several methods were quite comparable, though cosine similarity was decided upon.[2]

Initial excitement over non-negative matrix factorization's (NMF) potential was spurred by an application to facial image data[3] , where the basis images formed recognizable components of faces. However, subsequent applications to similar data sets did not fare nearly so well. It is now thought that the original result was due to a combination of unusually consistent image acquisition, and to the human facility for facial recognition. This led several researchers to look into ways to enforce additional constraints on NMF multiplicative update algorithms,[4] with mixed success.

III. ReADS - Unsupervised Learning

The Recurring Anomaly Detection System (ReADS) was developed in response to a need to identify recurring anomalies which may be indicators of larger systemic problems in a complex system. We define a recurring anomaly as any of the following:

1. A repetitive problem on either the same or different hardware,
2. A set of similar problems that are described in multiple problem reports, or
3. A set of problems that appear across subsystems or components but have the same root cause.

This section of the paper will describe the data and methods used, present the results, and discuss the capabilities and features of the ReADS tool.

III.A. Explanation of Data

A subset of the Shuttle Orbiter PRACA (Problem Reporting And Corrective Action) Data Support System (SOPDSS) dataset was used to analyze the recurring anomaly algorithms presented in this section. The full SOPDSS dataset is in use at NASA Johnson Space Center. The subset was randomly selected from a set of reports ranging in date from Jan 1, 1993 to March 31, 2003. The subset contains 333 reports. The text-based sections of all 333 reports were reviewed by subject matter experts. The experts determined that 70 of the reports fell into 20 identifiable recurring anomalies. The 20 recurring anomaly clusters contained anywhere from 2 to 10 reports. This is slightly under representative of the entire dataset in which 366 recurring anomaly clusters were identified each containing anywhere from two to 48 reports. But since over 40% of the clusters from the full dataset contain only 2 reports, it was decided that rather than forcing the larger recurring anomaly clusters to appear in the subset the random selection was reasonably representative. The text portion of the documents in this dataset are approximately 1-4 pages in length, and contain a large quantity of acronyms, abbreviations, part numbers, numeric values, and unusual punctuation, as compared to standard English text such as news reports. A PRACA report may be written over a period of time, by multiple authors with different expertises (e.g., technical, quality assurance, and safety evaluation). Many of the reports refer by name to other reports in the same dataset, and occasionally in other datasets.

III.B. Clustering and Anomaly Detection - Methods Used

The Recurring Anomaly Detection System (ReADS) takes advantage of two types of clustering:

1. Unsupervised clustering of the entire dataset using a modified version of the von Mises Fisher (vMF) algorithm;[2,5,6] and
2. Identification of higher priority clusters of reports which are possible recurring anomalies. These possible recurring anomalies are identified using two techniques: (i) document-document comparison using the cosine similarity measure algorithm,[6] and (ii) documents explicitly referencing other documents. This will be described in more detail below.

The reasoning behind implementing two distinct types of algorithms was the recognition that automated recurring anomaly analysis consists of two components: help the analysts more easily review all of the reports, e.g., unsupervised clustering of the dataset; and help the analysts prioritize their analysis by identifying the potentially high-risk reports, e.g., a subset of the entire dataset which are the potential recurring anomaly clusters. The recurring anomaly algorithms are run across the entire dataset, but as an analyst would expect, the majority of reports are not recurring anomalies.

There are an overwhelming quantity of reports to review. The breadth of topics covered in the reports is immense, ranging from out-of-range voltage readings to fuel injector leaks, and from incorrect laptop displays to meteoroid strikes. Therefore, the task of reading each and every report with the goal of associating reports with similar or related problems is a daunting one. The vMF clustering implementation clusters the entire dataset, thus providing analysts with the ability to select clusters of interest for further perusal. In this manner, analysts may still be required to read all of the reports, but they can focus their attention on a single group at a time. The second set of algorithms takes the next step by actually identifying possible

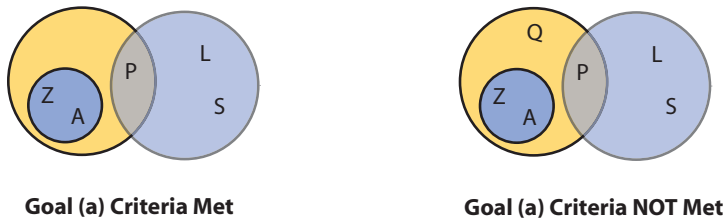

Figure 1. Let each letter represent a single report, and let each colored circle represent either expert or ReADS recurring anomaly clusters. Left figure: Objective (a) criteria met: One or more ReADS recurring anomaly clusters (blue) completely overlapping all of the reports in the experts recurring anomaly cluster (yellow). Right figure: Objective (a) criteria not met, report "Q" does not fall into any of the ReADS clusters.

recurring anomalies for the analyst. This provides the analyst with exactly what they are seeking. Therefore, the development of an automated system to extract recurring anomalies from text was considered worthwhile.

The recurring anomaly algorithm development was based on three objectives: (a) False negatives are unacceptable; (b) Match the subject matter experts' clusters; and (c) Possibly identify recurring anomalies missed by the experts. To meet the first objective, the ReADS recurring anomaly clusters only have to overlap completely with the experts clusters. Using a simple example, if an expert's recurring anomaly cluster contains three reports and those three reports fall into one, two or three ReADS recurring anomaly clusters, the objective is still met, see Figure 1. Of course, the case of identifying all reports as recurring anomalies is trivial and uninteresting.

The second objective is more challenging than it might initially seem. Individuals may not cluster reports in the same way as others, e.g., some will group smaller clusters together, others will not. And they may not even use the same measure of similarity for all of the reports that they cluster. The same cannot be said for the algorithms. The threshold defines the measure of similarity, thus the same level of similarity is used across the dataset. This disconnect results is a higher likelihood of overlapping the expert's results rather than matching the experts exactly. Since the tool is intended only to identify possible recurring anomalies, false positives are expected. Therefore if subject experts choose to break up, or combine, recurring anomaly clusters, this is considered acceptable. As illustrated in Figure 2, it is preferable for ReADS to overly combine, rather than overly separate expert clusters. Clusters that have been broken up by ReADS can

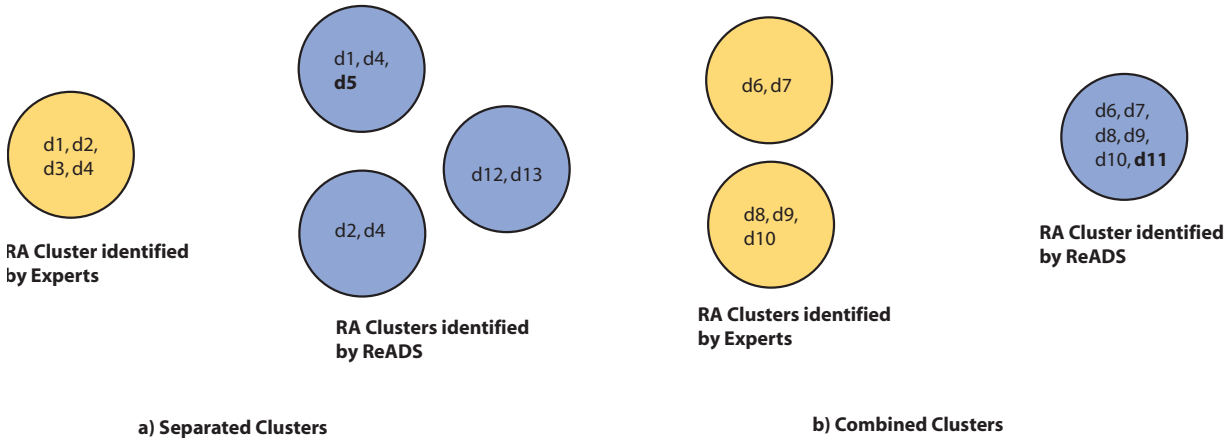

Figure 2. Illustration of separated and combined clusters.

be a problem. Instead of expecting experts to read and combine appropriate reports, this situation expects experts to read all of the reports categorized as recurring anomalies and combine appropriate clusters. Since recurring anomaly clusters are made up of reports, this situation contains nearly the same issues that the experts started with, save the ability to ignore all of the reports that do not make the full recurring anomaly list. Overly combined clusters, on the other hand, do not add additional complexity to the task. A subject

expert may choose to break up the bloated cluster, but in so doing, the expert was not required to hunt through any other clusters to decide how to break it up. All of the knowledge they need to make the decision exists within the one cluster.

The cosine similarity algorithm uses agglomerative clustering to identify reports which are similar within a set threshold. These reports are explicitly identified by ReADS as possible recurring anomalies. The approach is straightforward.

1. The dataset is reduced to a bag-of-words matrix. In this matrix, each row represents a single report and the columns represent the union of all unique terms in the dataset. The elements of the matrix are the word frequencies in each report.

2. Using a bag-of-words matrix, a document-to-document cosine similarity measure is calculated.

3. After calculating the distance between each document, the algorithm applies single linkage, i.e., nearest neighbor, to create a hierarchical tree representing connections between documents. Single linkage also generates an inconsistency coefficient which is a measure of the relative consistency of each link in the tree.[8]

4. The hierarchical tree is partitioned into clusters by setting a threshold on the inconsistency coefficient.

A high threshold on the inconsistency coefficient implies that the reports could be very different from each other and still be sorted into the same recurring anomaly cluster. Our preference has been to set the threshold low, thereby returning many smaller clusters of very similar reports. The threshold was determined qualitatively by comparing the results with the subject matter experts' results. There is no guarantee that this approach will determine an appropriate global threshold value across any assembled dataset. This is listed as future work. For completeness in the description of the algorithm, clusters consisting of only one document are excluded from the recurring anomaly results.

The document-referencing technique uses regular expression searches to identify reports which reference other reports by name. These referenced reports can be considered recurring anomalies, at least the author of the report believes that this is the case. Of course there could be a case where a report is referenced in the negative (e.g., "...report ABC is nothing like the problem we are experiencing..."). ReADS would incorrectly claim that the two reports are possible recurring anomalies, thus this is a false positive. But, as described below, the minimization of false positives is not the intent of this tool, nor is it required for the recurring anomaly detection task. Most problem report datasets provide authors a way to identify a given problem with other similar problems, and this approach takes advantage of that knowledge. The algorithm searches the document term list for matches to a regular expression. Once a match is identified, the algorithm confirms that a filename exists with the matched expression. Then the algorithm identifies the two reports as a recurring anomaly. The method has proven itself to be useful, and is such an obvious step to take, the ReADS tool would be lacking if it did not do so.

III.C. Clustering and Anomaly Detection - Results

Figure 3 shows the results of the cosine similarity component of the recurring anomaly detection system analyzed on the SOPDSS dataset discussed above. Items highlighted in light green are recurring anomaly clusters identified by ReADS using the cosine similarity technique which exactly match a cluster identified by subject experts. This is, of course, the ideal case and meets our stated second objective. Color-highlighted elements in the last row are recurring anomaly clusters identified by the cosine similarity technique in ReADS and not identified by the subject experts. These recurring anomaly clusters meet the third objective - identification of possible recurring anomalies missed by the experts. These clusters were handed over to a second set of subject experts who identified 8 of the 17 new clusters as legitimate recurring anomaly clusters, these are highlighted in pink. Six of the blue elements are recurring anomaly clusters which the secondary review deemed incorrect or trivial clusters. The remaining two blue elements were ReADS clusters where some of the reports in the cluster might qualify as a recurring anomaly, but the cluster also contained reports which did not qualify. The 8 newly minted recurring anomaly clusters meet the third objective and are examples of why this tool was developed. Four of the 8 newly identified recurring anomalies were discovered using only the cosine similarity technique. The other 4 were identified by both the cosine similarity measure and by the regular expression technique. The regular expression technique also solely identified even more possible recurring anomalies, but because this technique is not novel, these results are not included.

Shuttle CARS Toy Dataset		ReADS Recurring Anomaly Clusters																																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	RAs missed by ReADS
Subject Experts Recurring Anomaly Clusters	1											5																							
	2												2																						
	3																															2			
	4			4		1																													
	5														2																				
	6																2																		
	7																																		2
	8					2																													
	9																																		2
	10																										7								2
	11	4					2																												
	12																	3																	
	13						3																												
	14																																		2
	15																					1		5											1
	16		1											1																					
	17		1											1																					
	18																																		2
	19																																		2
	20																						7												1
	RAs missed by Experts		3		2	1		5	2	2	2					3			2	2	3	8			2	3	2	2	3	5	2		3	3	203

Figure 3. Confusion Matrix of subject expert identified recurring anomaly clusters vs. ReADS identified recurring anomaly clusters. RAs = Recurring Anomalies

Non-highlighted elements in the last row were identified by ReADS as possible recurring anomalies, but were not reviewed by the subject experts because a portion of the recurring anomaly cluster did match with one or more expert clusters, therefore it was decided that "extra" reports in a true cluster are acceptable. These extra reports may have value or they may not, but they cause little extra work for the analysts. Elements identified in light brown (the 34th column) are recurring anomaly clusters identified by experts but missed by the cosine similarity algorithm in ReADS, but which, luckily, were caught by the regular expression technique. The one report that was missed by both recurring anomaly identification algorithms (highlighted in purple) is a legitimate report in a recurring anomaly cluster, and it is not known at this time why it was missed.

The overall result of this analysis is that out of the 65 reports which fell into one of the 20 recurring anomaly clusters identified by the experts, ReADS caught all but 1, plus ReADS identified 13 more possible clusters. And this is the conservative perspective. A more liberal view of the results would state that out of 333 reviewed reports, ReADS caught all reports which belong in a recurring anomaly cluster, except 1 single report. And it is worthwhile to note that ReADS caught all of the other reports in that cluster. Plus, ReADS correctly identified at least eight other previously unidentified recurring anomaly clusters, containing a total of twenty-two reports.

III.D. ReADS - The Tool

The algorithms discussed above were integrated into an existing secure online search system which accessed several databases and provided search capabilities across the databases. When the users of this system complete their search and have the data that are of interest to them, they can choose to perform text mining on the reports. Due to the computational power and time it can take to perform text mining on large datasets, ReADS runs on local servers. Once the text mining is complete, the user is sent an email with a link back to the system to the location that their results are posted. The user has the option of viewing the results in two ways, tabular and visual. The tabular view has limitations, including ambiguity caused by reports referencing each other and therefore overly combining clusters. On the other hand, the visual representation developed using Tom Sawyer Software[a] nicely displays the data, using boxes to represent each report, and lines and arrows to indicate types of recurring anomalies. The visualization shows the vMF

[a] http://www.tomsawyer.com

clusters for the entire dataset, such as is shown in Figure 4. At this distant view, each report appears as a single point clustered with other reports. Using lines and arrows, ReADS also displays recurring anomaly clusters including the method of identification: referenced reports are connected with arrows; similar reports are connected with a solid line, groups of similar reports are assigned a hub to reduce the amount of lines drawn. Figure 5 is a closeup of one of the clusters of Figure 4. This visualization shows several things: similar reports, referenced reports, and a unique case of reports which reference reports which do not happen to be a part of the original dataset, i.e., the initial search did not contain the report in question.

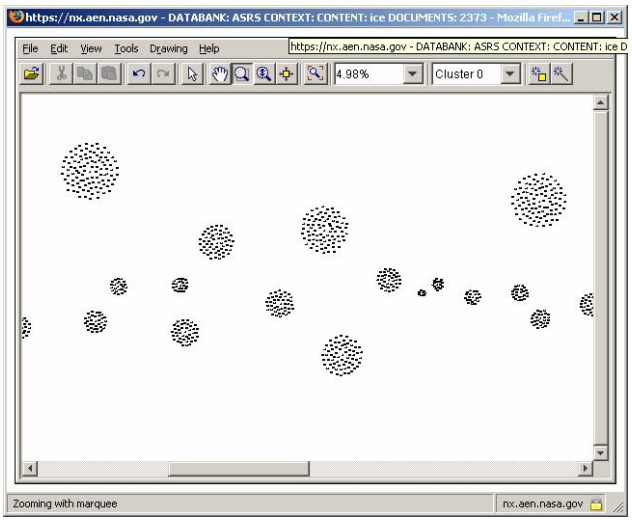

Figure 4. ReADS visualization of vMF clusters. Each small dot represents a single document.

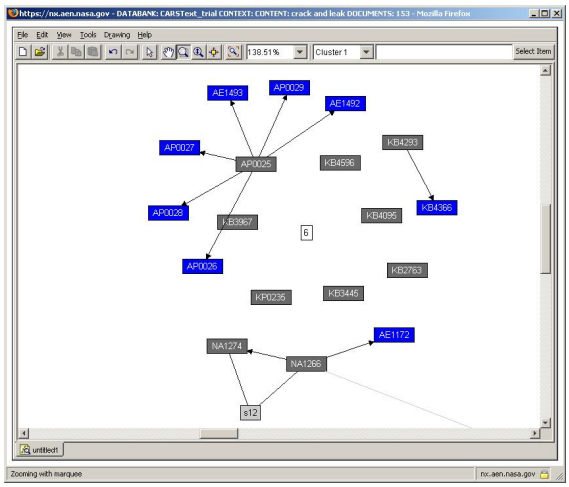

Figure 5. The closeup view of a vMF cluster shows several types of recurring anomaly clusters. Each colored box represents a single document

ReADS is a fully functional system. There are currently ten datasets available for text mining on ReADS. And the system provides the capability to search and text mine across multiple datasets, which opens up some interesting possibilities for discovering recurring anomalies.

IV. Mariana - Supervised Learning

The existence of a large set of preclassified data allowed for a different approach than the one used in ReADS. Supervised learning takes precategorized data, in this case narrative reports, as the learning dataset, and from those data build models which identify the distinctions between categories. The models are then used to categorize new, non-classified reports. There are many different supervised learning methods, several of which are discussed in this paper, and each have their own strengths and weaknesses.

There were two goals that we set out to achieve:

1. While acknowledging that there are no guarantees that the developed models will perform equally well on all datasets, our intent is still to build a generic tool with algorithms which have not been inherently designed to excel in the aviation domain.

2. A disadvantage of supervised learning techniques is that they are only as good as the learning data. Though the aviation report reviewers are experts and work hard to be consistent, fair and equable, they are human, and there is frequent disagreement between reviewers on the proper classification(s) of a report. Therefore, we decided that our minimum requirement would be to correctly classify a set of reports 75% of the time. Since the dataset is multi-category, categorizing a report into the wrong category is considered acceptable, as long as it is also categorized into the proper category.

IV.A. Classification of Safety Reports

The Aviation Safety Reporting System (ASRS) was created 30 years ago as a joint effort by the FAA and NASA.[1] The program was established to collect data and information about aviation events which could lead to unsafe situations, or non-standard procedures. This could be used to identify deficiencies and discrepancies in the National Aviation System so that appropriate solutions may be implemented. ASRS

reports are written by pilots, flight controllers, technicians and occasionally passengers. In general, the reports are filed in response to a specific event, but concerns and complaints are also filed. ASRS reports include factual information about the aircraft, location, parties involved and a narrative in which the author describes the event(s) and/or situation. These narratives are specific to aviation, filled with acronyms and abbreviations that only those with a working knowledge of the industry are likely to understand. Each report is read by at least two aviation experts who identify both hazardous conditions and underlying causes of the reported events. As of June 2006, there were over 600,000 documents in the ASRS data base with more than 3000 reports added each month. Many air carriers also have their own internal safety reporting system which may or may not be linked to the ASRS. The existence of these independent, or partially independent, systems led to the development of a master version of the categorizations which all organizations could reference. This version is known as the Distributed National ASAP Archive (DNAA).

Since this important dataset is growing at such a fast pace, human review will soon become the bottleneck in the system and therefore it is crucial to identify alternative techniques for identifying the potential safety hazards and causal factors. The field of information retrieval and statistical learning has developed many successful methods for classifying text and extracting subject information.[2] This has been a very active research topic in recent years and text classification algorithms are now commonly used in a wide variety of web and computer applications, such as search engines and spam filters. This section will describe how these methods were evaluated and modified to solve a unique problem in text mining. The outcome of this effort is an automatic classifier of aviation safety reports, known as Mariana. Mariana is a collection of tools to preprocess preclassified documents, build models (e.g., support vector machines or NMF) using those preclassifications, classify non-classified documents and present the results to the analyst. The system provides a quick, consistent and unbiased classification of the documents, thus improving the process by increasing the throughput, and reducing inter-operator differences.

IV.B. Multicategory and Sparse Data

Aviation and aerospace documents are quite different from existing data sets, such as Reuters-21578 and 20 Newsgroups.[18] Industry specific terms, references to equipment and positions, and the use of acronyms differentiate the reports from the text the public is commonly exposed to. The intended audience of aviation reports are expected to have a working knowledge of the industry so the reports are often void of descriptive or background material. The Reuters and 20 Newsgroups data sets are much more verbose and the verbiage is more representative of everyday common language.

To ensure the development and research of text clustering and categorization algorithms for the aeronautics domain we have built a data set specific to aviation safety. Since the Aviation Safety Reporting System (ASRS) narratives are a publicly available database, we identified a subset of the ASRS reports which are representative of the entire data set and used this subset for our analysis. The subset contained reports from 60 ASRS event categories. Srivastava et al. list several of these event categories.[2] The subset contained a few categories into which nearly half of the reports fall, as well as rare event categories in which experts would expect to see relatively few qualifying reports out of the entire dataset of 28,000+ reports. Due to the uneven distribution of reports, certain categories are overrepresented and can make classification of rare events difficult. A learning dataset was extracted in which 3,200 reports were categorized into 32 DNAA event categories. Due to the severely limited number of reports available in some of the rarer categories, our analysis focused on 22 of those 32 DNAA categories.

Table 1 demonstrates both the non-exclusive nature of the subject matter expert's assigned categories and the great imbalance between category populations, with ratios up to 37:1. Class 2 is assigned to better than half the narratives, classes 6 and 19 to more than a quarter each, while 12 of the 22 have populations under 6% of the total.

Table 2 highlights the exceedingly non-exclusive distribution of the subject matter expert's assigned DNAA categories. On average, the narratives are assigned to 2.16 categories, with assigned category counts ranging up to 10. Less than a third are assigned to single categories, more than a third have 2 categories, while the remaining third have 3 or more categories.

This dataset went through a text normalization and natural language processing program called PLADS. PLADS expands acronyms and collapses phrases to match differences of expressions with exactly the same meaning (e.g., 'FL' vs. 'flight level') and it sets nouns and verbs to the singular tense. This data set can be used for research purposes and can be downloaded from the NASA Ames Research Center's Intelligent Data

Table 1. Non-Exclusive Category Populations

Distribution of the set of 28596 ASRS documents, with respect to the 22 DNAA categories.

Cat#	#Docs	%Docs	-	Cat#	#Docs	%Docs
1	1876	6.56		12	4275	14.95
2	16173	56.56		13	2849	9.96
3	615	2.15		14	1654	5.78
4	610	2.13		15	508	1.78
5	3915	13.69		16	1249	4.37
6	7663	26.80		17	556	1.94
7	2230	7.80		18	1490	5.21
8	2844	9.95		19	8534	29.84
9	573	2.00		20	876	3.06
10	1456	5.09		21	441	1.54
11	514	1.80		22	807	2.82
				total	61708	215.78

Table 2. Non-Exclusive Categorization

Distribution of the full document set with respect to the number of categories per document.[a]

#Cats	#Docs	%Docs
1	9032	31.585
2	9926	34.711
3	6915	24.182
4	1911	6.683
5	565	1.976
6	154	0.539
7	68	0.238
8	16	0.056
9	8	0.028
10	1	0.004

[a]With 28,596 documents and 22 categories.

Understanding group website.[b]

Table 3 illustrates the unusually severe asymmetry of the PLADS processed ASRS narrative's term occurrences. Fully one third of the extracted terms are only found in single documents, and so are completely useless for statistical text mining. Similarly, terms that are only found in small numbers of documents can provide little information about the set. Fully 90% of terms could be eliminated while retaining 90% of the counts and the count sums, which constitute the information used in statistical text mining. On the other hand, almost 1% of the terms are each found in over 14% of the documents. Unless their distributions have strong statistical properties, these terms also provide little information, and so become candidates for an ASRS domain specific stopword list.

Figure 6 illustrates the 22 DNAA categories' binary population correlations, i.e., the relative degree of co-occurrence. The strongest correlations, which are still quite weak, involve the three largest population categories. The largest correlations involve categories 2, 6, and 10 with correlations ranging from 0.47 to 0.33. There are also a fair number of 3-way correlations, but the number of significant higher level correlations decreases rapidly. High correlations between large and small population categories would imply that essentially all instances assigned to the smaller category are also assigned to the larger.

Since there are no obviously strong correlations between the different categories, we assumed category independence. So, instead of building one large multi-category classifier we treated each category individually and set up a series of binary (e.g., 'in the class' or 'out of the class') classification problems. This gave us some extra flexibility, but by treating each report as being either "in" or "out" of a category we added to the imbalance problem of the data by increasing the number of reports considered "out" of each category.

All of these factors work against the direct applicability of most conventional statistical text mining techniques, which have largely been developed for data sets having significantly different statistical properties. This was the primary motivation for this work.

IV.C. Dimension Reduction

In order to improve classification rates, dimension (i.e., term) reduction is often used to eliminate terms that add "noise" to documents. These terms can be terms that occur very frequently or very rarely and are not associated with particular categories. Human readers are able to ignore these terms and identify the relevant information to extract concepts and infer meaning extremely effectively, but it is much more difficult to train

[b]http://datamining.arc.nasa.gov/

Table 3. Assymetry in Term Distributions

Distribution of the dataset with stopword filtered 2+character terms with respect to the minimum number of documents where term occurs.[a]

#Doc	#Term	%Term	#Count	%Count	Sums	%Sum
1	25729	100.00	1714931	100.00	2615965	100.00
2	17603	68.42	1706805	99.53	2605072	99.58
4	12560	48.82	1694903	98.83	2588504	98.95
8	8991	34.94	1676237	97.74	2563263	97.99
16	6103	23.72	1644694	95.90	2521891	96.40
32	4065	15.80	1599497	93.27	2464005	94.19
64	2685	10.44	1537551	89.66	2384040	91.13
128	1731	6.73	1452340	84.69	2273663	86.91
256	1068	4.15	1331842	77.66	2113264	80.78
512	609	2.37	1164957	67.93	1891332	72.30
1024	351	1.36	978935	57.08	1638920	62.65
2048	162	0.63	707972	41.28	1267763	48.46
4096	69	0.27	445444	25.97	868293	33.19
8192	14	0.05	151219	8.82	364014	13.91
16384	1	0.00	17070	1.00	51928	1.98

[a]With 25,729 terms from 28,596 documents.

a computer to do the same. Preprocessing of the data is often necessary to reduce the number of terms to only the most informative.

Simple statistical methods like Chi-Squared, TFiDF, and Information Gain can be used to find a small set of terms that describe a single category, but because of the restricted vocabulary used within the field of aviation and the limited number of terms within each document these sets often overlap many, if not all, of the categories, i.e., very similar terms can describe more than one category.[19] It is often a combination of terms and common use words that are the most informative within the document.

An alternative approach to the methods mentioned above is to use Natural Language Processing (NLP) to extract phrases and sets of terms to help reduce the noise within the text and, thus, improve classification. This is a very effective method but most NLP systems are rule based and require a significant amount of effort to build and evaluate. For specialized documents like Aviation Safety Reports this would also require subject expert involvement. Since our first goal is to build a classifier that is robust against the the common language variations seen in the reports and requires minimal maintenance and human involvement to build, we decided against using NLP and instead focused our attention on classification methods.

IV.D. Support Vector Machine (SVM)

To develop a system which would require little interaction, we looked at several multicategory techniques such as Naive-Bayes, Neural Networks, ADABOOST, and SVM as well as several other methods.[6] Both raw text and PLADS preprocessed text was compared. The terms were ranked by Information Gain and the bag of words matrix was reduced to the top 500, 1000, 2000, and 5000 terms. To evaluate the performance of the classifiers we used the area under the ROC curve.

As can be seen in Figure 7, standard SVM consistently matched or outperformed the other methods across all categories. SVM also gave us some flexibility for improving the results and adapting the algorithm to match the data set being classified. Recently, comparable results were demonstrated using non-negative matrix factorization (NMF). In this paper we present our advancements of both SVM and NMF and demonstrate our ability to automatically categorize aviation safety reports.

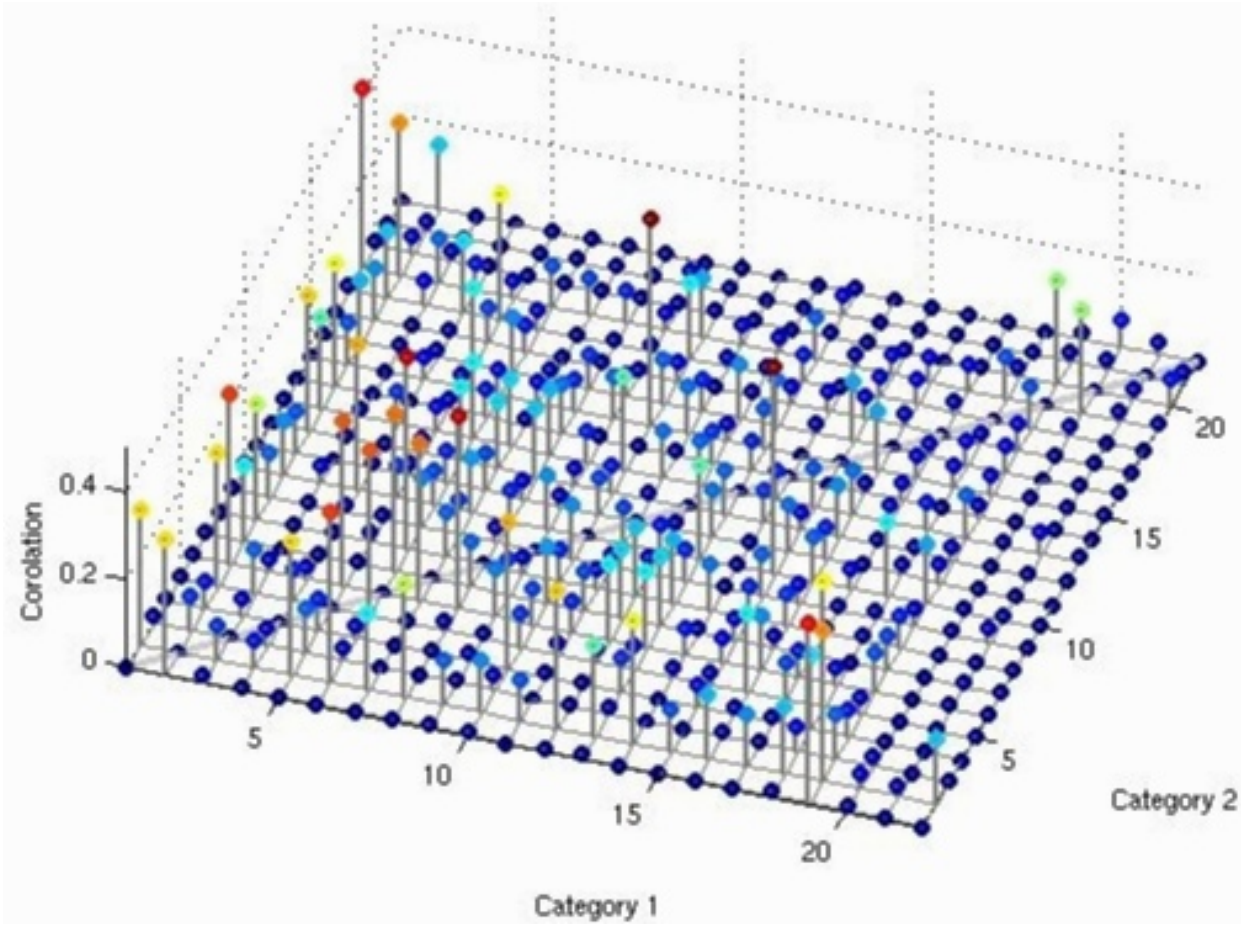

Figure 6. Binary correlations of the dataset categories. Computed as $c_{ij} = (n_{ij}^2/(n_i * n_j))^{1/2}$, for n_i, n_j, and n_{ij} the respective populations. Diagonal elements, all with value 1.0, have been suppressed.

Support Vector Machines were initially developed by Vladimir Vapnik in 1979.[17] SVMs are a statistical learning method based on Structural Risk Minimization (SRM). In SRM, we choose a set of classification functions or hypotheses that classifies a set of data. A risk functional measures the expected error rate for the set of classification functions. By minimizing this functional we are putting bounds on the empirical risk.[17]

The simplest example of a Support Vector Machine is a linear hyperplane trained on data that is perfectly separable as shown in Figure 8. Given a set of input vectors, $\mathbf{x}_i \in \mathbf{R^d}$, and labels, $y_i \in \{-1, 1\}$, SVM finds a hyperplane described by its normal vector, $\mathbf{w}$, and distance from the origin, b, that divides the data perfectly and is equidistant from one point in each class that is closest to the hyperplane. This hyperplane is a decision boundary and the classification of an unknown input vector is determined by the sign of the vector operation

$$\mathbf{x}_i \cdot \mathbf{w} - b = d \tag{1}$$

If $d \geq 0$ ($d < 0$) then the input is likely in the class $y = +1$ ($y = -1$).

If the data is not perfectly separable this method can be adapted to compensate for instances that occur on the wrong side of the hyperplane. In that case a slack variable, ξ, is introduced that measures the error of misclassified instances of the training data. SVMs find a hyperplane that best separates the data and minimizes the error, ξ.

$$V(\mathbf{w}, b, \xi) = min(\| \mathbf{w} \| + C \sum \xi) \tag{2}$$

where C is a user defined parameter of the slack variable. If C is large, the classifier puts a large penalty on errors. If C is small, then the classifier is more general. If the number of inputs in and out of the class are not balanced, as is the case for the aviation safety reports, it is sometimes necessary to factor the errors of

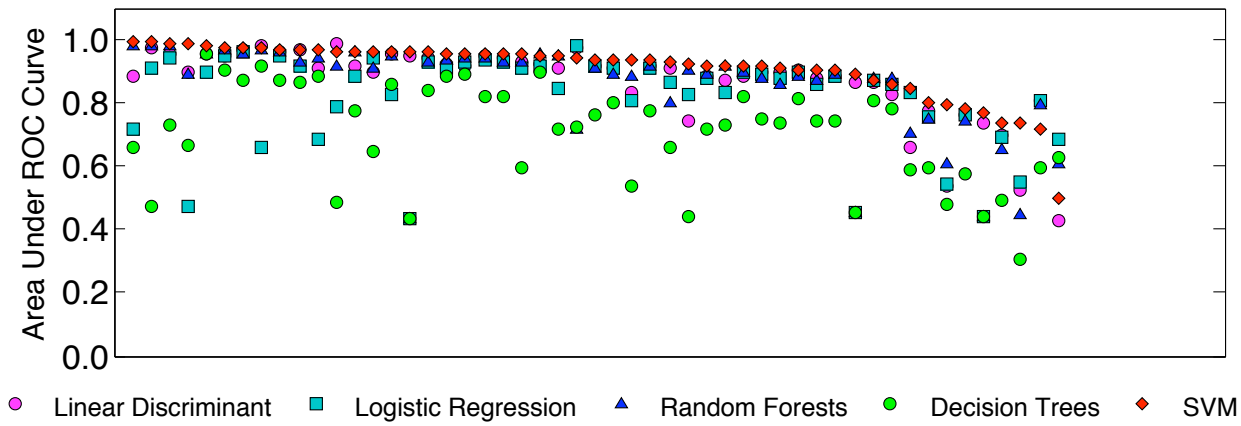

Figure 7. Performance of different classification algorithms on 60 ASRS categories

one class more than the other. An additional parameter, μ, is added to weight one class error over another.

$$V(\mathbf{w}, b, \xi) = min(\| \mathbf{w} \| + \mu_{y_i} C \sum \xi) \tag{3}$$

The decision hyperplane can only be linear which is not suitable for many types of data. To overcome this problem a kernel function can be used to map the data into a higher or infinite dimensional space and run SVM on this new space.

$$\Phi : \mathbf{R} \mapsto \mathcal{H} \tag{4}$$

There are many kernel functions available and new kernels can be built from the data itself. The kernels only need to meet Mercer's Conditions to be used in SVM.[16] For our classifier we chose a radial basis function,

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2} \tag{5}$$

where γ is a parameter input by the user.

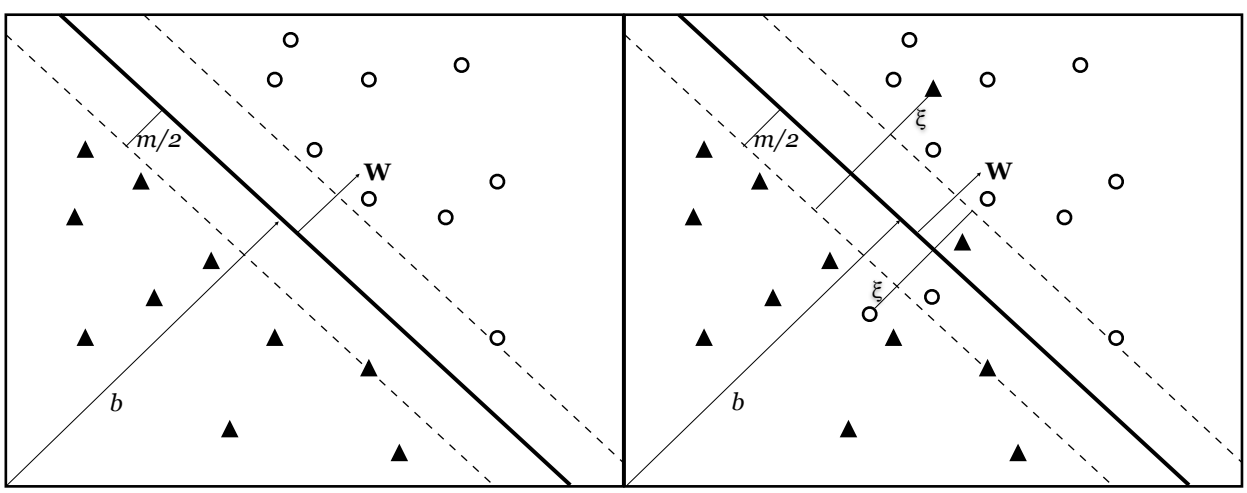

Figure 8. Optimum Hyperplane for Separable and Non-Separable case.

IV.D.1. Hyperparameters

For the set of aviation safety documents, we have found that the values of the hyperparameters, C, μ, and γ, greatly affect the performance of the classifier. The number of dimensions or terms, unknown relationships between categories, and the different ratios of documents in each class make it difficult to build a general set of parameters. So, they are either determined from basic tests on the input data or found by a simple grid

search.[14] In practice, C and γ are determined by starting with a course mesh and repeating several passes at increasingly finer scales. This can take hundreds of evaluations for just two hyperparameters. When adding the μ parameter to compensate for the unbalanced data, it can take thousands of iterations to complete even a coarse grid. Instead of using a brute force grid, we used a Simulated Annealing search to determine the optimum hyper-parameters.

IV.D.2. Simulated Annealing

Simulated Annealing is a stochastic search optimization similar to Markov Chain Monte Carlo.[15] The basic algorithm evaluates the output of the classifier for a given set of parameters and then randomly adjusts the values of the hyperparameters for reevaluation. If the new set of hyperparameters improves the results, the new set is kept. If it does not, then the results are kept or rejected based on the probability of how different the current results are from the best results thus far. Modifying the hyperparameters even when they are moving away from the current maximum reduces the probability of settling on a local maximum, thus the final result is likely the global maximum. The data set is broken into three parts, 50% for training the models, 25% validation to check the performance of the models, and 25% as a final test. The natural distribution of the categories is maintained in each part. Since we are treating the multicategory problem as a set of binary classification problems, there are significantly fewer documents in each class than not in the class. To ensure that there are enough positive examples, the training file for each binary classification is built by randomly selecting document vectors from the training set while maintaining a minimum 10% documents that are in the class. The validation and test set maintain the natural percentage of the entire corpus of documents. Once a model is built from the training set, it is used to predict the categories of the validation set. We use the area under the ROC curve to evaluate the predictions. This process is repeated until a maximum is found or for a predefined number of iterations. Once the optimum hyperparameters are found for each class the model is then used for a final prediction of the validation data.

IV.D.3. Confidences

The outputs from the prediction is the distance from the hyperplane and can be any real number. This can be misleading when comparing the outputs from different class models for the same document. It is natural to assume that a larger number is the more correct answer, but it is not possible to directly compare the output of two different categories. Even if the outputs are comparable in range, the distance is not a true measure of confidence in the prediction. The categorization is determined by the sign of the output from the SVM and it traditionally discards the distance. We combine the distance and the rates that each category occurs in the data set to generate a model of the confidences,

$$Conf = \frac{1}{1 - exp(-\alpha d + \beta)} \tag{6}$$

where α and β contain category information. The final output is a positive number between 0 and 1 and scaled; so anything above 0.5 is considered in the class.

IV.E. SVM Results

The optimum choice of hyperparameters improves the area under the ROC curve by 10% or more in some of the categories on raw text. It also performs equally on text that has been NLP preprocessed by the PLADS system. As shown in Figure 9, there is little difference in performance when this process is used on raw text and PLADS processed text, without the overhead NLP techniques tend to require.

Once convinced that the algorithms were comparing well against other techniques, we evaluated the performance of the full tool by auto-classifying 100 randomly selected reports and having the results reviewed by a problem report expert. The reviewed results were encouraging. Our stated goal was for the reviewer to approve of at least one of the auto-classification selections 75% of the time. The reviewer agreed with the top classification 73% of the time. The reviewer agreed with one or both of the top two classifications 86% of the time, and with the top three classifications 90% of the time. A separate review of the 100 reports was done by another subject expert. Then our reviewer reviewed their classifications, and agreed with their top (and only) classification 89% of the time. These results not only indicate that Mariana is well within expected classification levels, but that reviewers disagree amongst themselves. Since there is no expectation that an

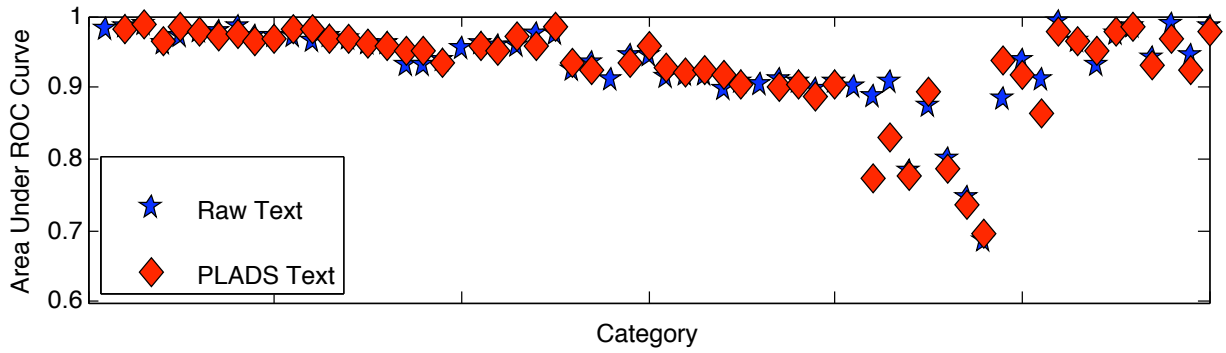

Figure 9. Comparison of Raw Text and PLADS Processed Text using Optimum Hyperparameters

automated learning system could do better than the human classifying that it learns from, the Mariana results are considered quite positive.

IV.F. NMF for ASRS Text Categorization

Non-negative Matrix Factorization is a variation on the host of mathematically motivated techniques for factoring large vector valued data arrays into basis and distribution matrices. The general approach is to seek a relatively small set of basis vectors, W, and a corresponding set of distribution weight vectors, H, such that data, $X \approx W * H$, by minimizing some measure of the difference, $X - W * H$. The hope is that the basis vectors will correspond to some fundamental properties of the data set, with the distributions mapping those properties to the data.

In NMF applications, the data values are non-negative, typically counts or scalar measurements, and the factorization is constrained to keep both W and H non-negative. This has a strong appeal lacking in factorization methods that allow either basis or distribution values to go negative. With non-negativity, the basis may be thought of as components, and the distributions as recipes for adding components to match the data. Thus NMF overlaps the traditional field of statistical mixture modeling.

While non-negativity is an appealing property for factor-based data mining, constraining conventional difference minimization algorithms to maintain non-negativity can be a serious complication. This changed circa 2002, with spreading recognition of the potential of Lee & Seung's multiplicative update approach.[3,7,9] For the squared Frobenius norm, the standard sum of squared matrix values, Lee & Seung's original paper gives the minimizing reestimation relations as:

$$W_{ab} = W_{ab} \frac{(X * H^T)_{ab}}{(W * H * H^T)_{ab}} \tag{7}$$

$$H_{bi} = H_{bi} \frac{(W^T * X)_{bi}}{(W^T * W * H)_{bi}} \tag{8}$$

where a and i index over the attributes and instances of X, respectively, and b indexes over the basis vectors. This is actually a reformulation of the standard gradient driven norm minimizing search, augmented with a conceptually simple step size computation that maintains the non-negativity constraint.

Starting with non-negative W and H, and applied alternately, these reestimation equations are proven to monotonically lower the norm toward a local stationary point, while maintaining the non-negative properties of W and H. For one accustomed to the complexities of conventional constrained function minimization algorithms, this result is altogether remarkable. Lee & Seung also provided an alternate multiplicative minimization for the Kullback-Liebler divergence of probability matrices. Dhillon & Sra,[10] and Wang et al.[11] have since developed versions for other matrix norms. Other researchers have sought ways to enforce additional constraints, with mixed success.[4]

IV.F.1. NMF Practicalities

While NMF is a potentially powerful tool for factor based data mining, it is not one that can be applied blindly with any expectation of useful results. Every data mining project involves considerations of data

specification, collection and preparation that will affect the results of any analytic method. The factorization only provides a compression and reorganization that reflects the dominant statistical properties of the data as presented. Data understanding depends on disambiguating the resulting basis and distribution, usually in the light of much additional information.

Preprocessing considerations involve choices for data specification, acquisition, selection and preparation. Data must be informative with respect to the focus of the analysis. Failing this, irrelevant aspects will very likely dominate the results, and will certainly degrade sensitivity to the aspects of interest. Factor based methods are intrinsically intolerant of unknown values. Thus incomplete data instances must either be discarded or repaired. Likewise, data weighting to emphasize various statistical properties is subject to ongoing research, with demonstrated but still largely unpredictable consequences for end results. Regardless of the chosen analytic method, careful thought is needed to identify how any specific data set could be handled to maximize end utility, and considerable experimentation to confirm what works and how well.

In factoring the prepared data, there is the fundamental choice of what difference matrix norm shall be minimized, with an expanding range of choices, and consequences that are not yet well understood. A basis size must be chosen, or a range of sizes searched over and evaluated. Algorithmic details, particularly factor initializations, may have significant effects. Since the multiplicative NMF algorithms are gradient driven, they approach their stationary points at exponentially decreasing rates. This requires stopping criteria that must balance the opposing requirements of computational efficiency and numerical accuracy.

Once the data has been factored, understanding the basis and distribution can be challenging. This is a major problem for undirected data mining. Understanding is largely a matter of relating results to external criteria, and so requires additional information, e.g., subject expertise. In supervised data mining we have the advantage of a standard against which we can compare our results and evaluate our algorithms. Either way, making sense of the factorization entails another level of algorithmic decisions that affect both our understanding and our confidence in that understanding.

Since the focus of this effort has been to use statistical text mining techniques, bag of words matrices are the mathematical basis of the work. Thus, the specified data consists of a count for each term occurrence in a document. Settling on a definition for the word 'term' can be managed on a per database case. It is customary to discard a set of 'stop words', terms thought to be too common to provide any useful information. Our current system uses the standard English stopword list[c]. Terms unique to any single document are clearly useless, although surprisingly common in the ASRS narratives. A variety of term count weighting schemes have been devised, intended to emphasize some aspect of the statistics. We have found that term weightings which improve final product quality may also greatly increase the error norm that our NMF algorithm minimizes, and vice versa, so end-to-end testing is essential.

For our applications, we seek to associate documents with pre-determined categories, by using manually categorized documents for training the system. This gives us a category count, and so a minimum basis size, though this does not necessarily provide enough data on some of the rarer categories. And while manually assigned categories provide an objective measure of factorization utility, their use requires an auxiliary algorithms to generate categories from our factorizations.

IV.F.2. NMF Application

The work described here was inspired by the promising results shown in the Allen et al.[12] paper which came in second place in the 2007 SIAM Text Mining Competition. This effort is a follow on of that entry, aimed first at reproducing their results, and then to explore the effects of alternatives in weighting, factorization and classifier construction. Our classifier construction code is a Matlab memory limit adaptive matrix coded reimplementation of that submitted by Allen et al. Predictive quality is evaluated per the SIAM competition specification,[13] basically a sum of the traditional receiver operator characteristic (ROC) curve area and a category confidence measure. Using these, we have duplicated Allen et al.'s results, improved on that result by approximately 25%, and investigated a range of variations on their approach to NMF based tutored classification.

Allen et al. used their General Text Parser to extract terms, accepting any strings of 2 to 200 characters, which appear in 2 or more reports. We used our own term extraction code, with both 2- and 3-character minima. While placing no limit on term length, we found none that exceeded 200 characters. Both of us used SMART's English stoplist. We kept all terms and subsequently sifted them for several levels of minimum

[c]SMART's English stoplist at ftp://ftp.ce.cornell.edu/pub/smart/english.stop

document number occurrences, getting 15,431 2-character and 2 document minimum terms, for Allen et al.'s 15,722 terms.

We generally used Allen et al.'s entropy based term weighting. For each term t, they compute an entropy with respect to the document set as $g(t) = 1 + \sum_d (p(t,d) * log_2(p(t,d)))/log_2(nD)$, where $p(t,d)$ is the observed proportion of term t in document d, over the nD documents. Thus $g(t)$ ranges from 1, for a term appearing in a single document, to 0, for one that is uniformly distributed over all documents. Their final weight for occurrence count$X(t,d)$ is $g(t) * log_2(1 + X(t,d))$, thus de-weighting large occurrence counts. No within document normalization was applied.

We briefly investigated two simple alternative term weightings. Using the raw unweighted term counts reduces the NMF approximation error norm about 30 fold, relative to the entropic weighted terms, but degrades our final prediction quality. Using raw counts with stop words retained gives an additional 3 fold reduction in NMF error norm, and significant reduction in prediction quality. Thus the entropic weighting clearly amplifies the statistical properties pertinent to our categories, despite the increased cost in the factorization objective function.

Very limited experiments with 3-character minimum terms, at several minimum document number levels, show no significant degradation of predictor quality. This was unexpected, and is thought to be due to the PLADS processing of the ASRS reports, which tends to generate long character sequences. We find only 219 2-character terms, when sifted at the level of minimum 2 documents per term.

We investigated the effect of moderate selection for term distribution, sifting the full term set for those having minimum document counts of 2, 5, 10, 20 and 40. See Table 3. As alluded to in a previous section, at a minimum of 40 documents per term, we eliminate about 90% of the original terms, while retaining about 90% of the original term occurrence counts. Thus there are clear computational advantages to term sifting. We find a clear response to term sifting in the NMF approximation error, with a minimum squared error norm consistently observed at 10 documents minimum per term. This response at the NMF level does not extend through to the quality of our predictions. Seemingly random prediction quality variations, between otherwise identical runs, and attributed to the random NMF factor initialization, generally exceed the variation between the document count groups' mean quality. Group mean predictive quality increases roughly as the log of minimum document count, with a total increase of 2-3% over the range from 2 to 40, with about 5% within group variation.

Both of us used the original Lee & Seung algorithm for minimizing the summed squared approximation difference, $W * H - X$, starting from random W and H matrices. Our NMF code has evolved from the public distribution [d] of Dr. Patrick Hoyer,[4] but was run in the same basic mode that Allen et al. ran theirs, with identical core computations.

Allen et al. used an NMF basis size nB of 40, almost twice the 22 categories provided, with only 5 convergence cycles for the factorization. We investigated the effect of varying the NMF basis size, choosing nB of 33, 44, 55 and 99 as reasonable given the category count of 22. At these levels, increasing basis size gives a small but consistent decrease in the error matrix norm and a corresponding increase of a few percent in the prediction quality, with much larger increase in computational costs. This effect is independent of the response to minimum document number.

We investigated the effect of increasing the number of NMF convergence cycles above the 5 used by Allen et al., and this is the source of much of our improvement over their results. Extending the convergence cycles from 5 to 20 gives about a 20% increase in predictive quality, with a slight further increase out to 50 cycles, and insignificant gains beyond that.

The summed square NMF approximation error trace has always shown the exponentially decreasing rate characteristic of difference driven algorithms. We found none of the rate variations suggestive of complex objective function topography. This may be due to the search space's large dimensionality. On the other hand, it is quite clear that we are finding local minima in the objective function surface, and that these minima generate significant variation in category predictive quality. Thus it is better to do multiple short convergences and choose the best, rather than a few long ones, so long as random initialization is used.

The predictor construction used by Allen et al. is fundamentally

$$C_p = f(H_p^T * (H_t * C_t)) \tag{9}$$

where inputs, H_t and H_p, are the NMF training and predictive distribution matrices, C_t and C_p, are binary valued non-exclusive training and desired predictive document category indicator matrices, and $*$ denotes

[d] http://www.cs.helsinki.fi/patrik.hoyer/

matrix multiplication. Function, f, combines a level cutoff and binary conversion from assessments to logical predictions. In practice, the H_t and H_p are both filtered for factorization noise and normalized, while the $(H_t * C_t)$ requires a further normalization w.r.t. H_t.

There is a clear interaction between the degree of NMF convergence and the optimum noise filter level applied to the distribution matrices, H_t and H_p. As expected, limited convergence requires tight filtering, while better convergence permits looser filtering. The gain in predictive quality due to the optimum parameter shift is small relative to that due to the better factorization. The need for this filtering is largely due to their conversion of H_t to a binary indicator matrix, prior to computing $(H_t * C_t)$. We find, for well converged factorizations, that the filtering can be eliminated if H_t is simply normalized. This is a consequence of the computational noise introduced by the binary conversion.

Symmetry suggests that $H_p * C_p = H_t * C_t$ should give better predictions than equation 9, preferably as $C_p = H_p^{-1} * H_t * C_t$. Unfortunately, H_t is very far from being invertible. We are investigating ways to solve this symmetric form. The best result obtained thus far uses the H update of the Lee & Seung NMF algorithm for $W * H = X$, to compute C_p from known H_p and $H_t * C_t$. The resulting predictor qualities are only a few percent better than what we get with equation 9, and obtained at considerably greater computational cost.

Allen et al.'s entropic term weight $g(t)$ sums $p(t,d)log_2(p(t,d))$ over all documents d. This works well when all data are at hand and can be processed together. In a production system it will be advantageous to train the system on initial data, and then apply it to new instances as they become available. It is then necessary to retain $g(t)$, possibly update it, and apply it to new instances. We have not investigated how $g(t)$ might evolve in such circumstances, using the fixed value from the training data for our production mode experiments.

Getting an H_p in production mode is more difficult than getting $g(t)$. Allen et al. factored the joint training plus test data set, as $W * (H_t, H_p) = (X_t, X_p)$, separating H_t and H_p afterwards. This suggests using $W_t * H_p = X_p$, which can be solved using a single sided Lee & Seung convergence code, after selecting W_t and X_p for common terms. The predictive quality is only slightly lower than that obtained with joint factoring. This is as expected, on account of the smaller data set used to generate W_t, which only has 3/4 the document set and 90% of the terms available in the joint calculation. There is a potential savings in computational efforet, which we have not attempted to quantify.

IV.G. Mariana - The Tool

Now that the algorithms behind the supervised learning problem have been explored, this section will describe the tool that was developed to support the analysis of the DNAA and ASRS reports. We have built an interactive web tool that incorporates the SVM models and classifier with an online database of ASRS reports. The flowchart in Figure 10 describes the process.

The subject expert analysts can log onto the website and select a set of reports to be auto-classified. Once a set of reports are selected, Mariana will predict the categories the documents belong to and display only the predicted categories that have a 50% confidence or more. The user can then evaluate the report and confirm or correct the classification. Figure 11 is a screenshot of the tool displaying a sample report, and the results of the Mariana classification algorithm. And, in fact, as can be seen in the figure, the classifications are ranked by confidence in the prediction. The Mariana auto-classification online system will start a six-month trial at an air carrier's site in May 2007.

The reports can be displayed to an unlimited number of reviewers and each of the expert's classifications are recorded. Thus providing an ability to track both the performance of the auto-classification models and algorithm, and the inter-operator error. New models can be easily built, using SVM or NMF, when new data is available or the models' performance drops.

V. Future Work

There are two topics being evaluated to improve ReADS. The first was briefly mentioned earlier in the paper. The recurring anomaly cut-off threshold could be quite different for different datasets, and that is not being addressed in the current version of ReADS. Research needs to be done to either identify a global threshold for determining recurring anomalies, or to develop a technique which self-determines the threshold. The second issue arose early in the design and development of ReADS, and that was the need to handle very large datasets (150,000+ documents). The vMF clustering algorithm scales reasonably well, but the cosine

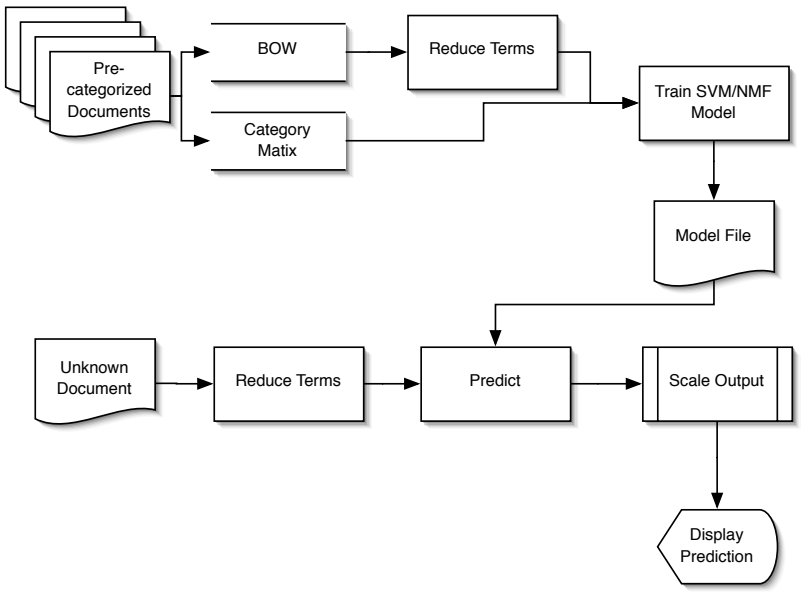

Figure 10. ASAP Classification Flow in Mariana

similarity algorithm does not. This has led us to contemplate using the vMF results to help identify the non-recurring anomaly documents, thereby reducing the amount of document-to-document comparison required to identify recurring anomalies. The approach would take advantage of the ranking of the vMF clustering algorithm to eliminate a significant portion of the documents prior to running the document-to-document comparison. This might allow us to analyze larger datasets without running into computation limitations. Early work is already showing some promising results.

The supervised learning effort also has two SVM topics that will be addressed in the near future. The SVM method will soon be applied to the more challenging problem of addressing a more subtle classification problem in the aviation industry. Each of the categories discussed in this paper have been further broken up into sub-categories. Models will be developed to categorize aviation safety reports not just into the top-level category but also the proper sub-categories. It needs to be seen whether the SVM method performs well on the sub-category level. The second topic will be to perform a post-trial evaluation of the Mariana's performance at the air carrier site. It is likely that integration of non-narrative information available in the aviation reports will be a high priority, with the expectation that this information will bring up the classification rates on some of the rarer categories.

Since the NMF results showed promise, the non-negative matrix factorization work will also continue. The largest single factor affecting prediction quality is the entropic weighting. It is not entirely clear why this is true, and this needs further study. The NMF literature includes a number of alternate weightings that should also be considered. In our investigation, the Allen et al. prediction generating algorithm proved surprisingly effective. It is not yet clear if this is due to efficient use of information in the generator, or if the factorization basis is a close match to the ASRS categories. This too needs further investigation.

Allen et al.'s entropic term weight $g(t)$ sums $p(t,d)log_2(p(t,d))$ over all documents d. This works well when all data are at hand, as in the contest situation. For a production system, such as Mariana's deployment at the air carrier site, it will be advantageous to train the system on initial data, and then apply it to new instances as they become available. It is then necessary to retain $g(t)$, possibly update it, and apply it to new instances. We have not investigated this evolution of $g(t)$, using the fixed value from the training data for our production mode experiments.

VI. Conclusion

Identifying recurring anomalies in large datasets is challenging and fraught with the possibility of missing a set of related reports, a tool, such as ReADS, provides insight into possible recurring anomalies using

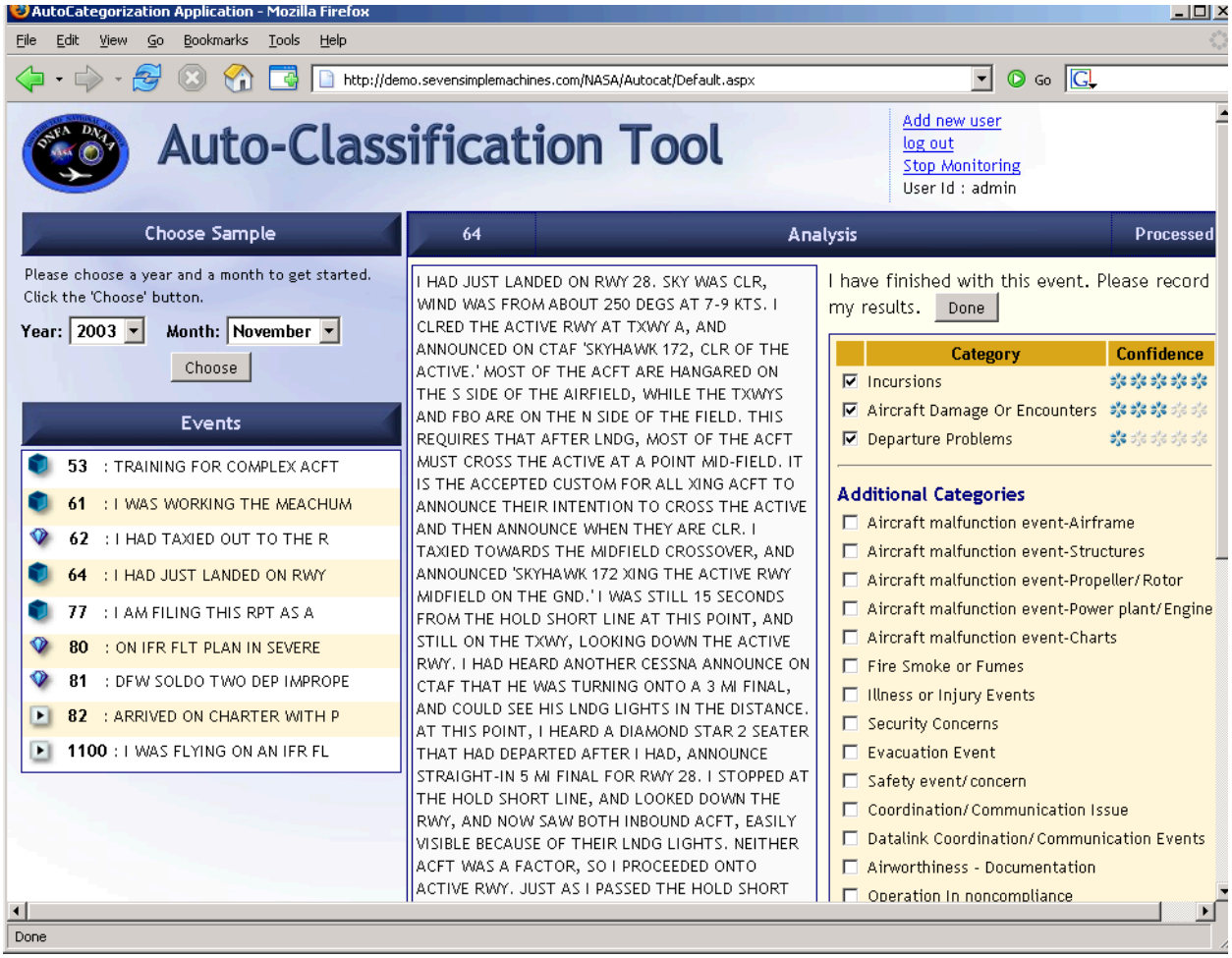

Figure 11. Screenshot of Mariana

statistical methods. And, importantly, provides analysts with a way to narrow their focus and attention to the highest priorities, those selected by ReADS as possible recurring anomalies, and then provide the same analysts with a way to divide up all of the data, i.e., clusters, so they can approach the reports in a more systematical way.

For the aviation classification problem, both SVM and NMF are compelling approaches. The SVM method with optimized hyper-parameters results in high quality predictions, and there are techniques which can be applied to improve it even further. The simulated annealing method for determining the optimal hyper-parameters allows for class specific error rates which is one of the strengths of the described approach. Non-negative matrix factorization for ASRS document categorization, using the standard Lee & Seung NMF algorithm, with entropic term weighting and the Allen et al. prediction generation algorithm, has also produced remarkably consistent high quality predictions. This holds over a fair range of basis sizes and minimum document count levels, and over a large set of factorizations, where between type quality variation is comparable to within type initialization noise. Major loss of prediction quality occurred only when the entropic weighting was omitted, when the factorization was insufficiently converged, or when low probability assessments are accepted as predictions.

References

[1]NASA, "Aviation Safety Reporting System (ASRS) Program Overview", 2007: http://asrs.arc.nasa.gov/overview.htm.

[2]A.N. Srivastava and B. Zane-Ulman, "Discovering Recurring Anomalies in Text Reports Regarding Complex Space Systems," *Proceedings of the 2005 IEEE Aerospace Conference*, 2005.

[3]D.D. Lee and H.S.Seung, "Learning the parts of objects by non-negative matrix factorization.", *Nature*, 401(6755):788-791, 1999.

[4]P. Hoyer, "Non-negative matrix factorization with sparseness constraints", *Journal of Machine Learning Research* 5:1457-1469, 2004.

[5]A. Banerjee, et al., "Generative Model-based Clustering of Directional Data," *SIGKDD '03*, Washington, DC, USA, August 2003.

[6]A.N. Srivastava, et al., "Enabling the Discovery of Recurring Anomalies in Aerospace Proplem Reports using High-Dimensional Clustering Techniques," *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, March 4-11, 2006.

[7]D.D. Lee and H.S.Seung, "Algorithms for Non-negative Matrix Factorization", *Advances in Neural Information Processing 13* (Proc. NIPS 2000), MIT Press, 2001.

[8]The MathWorks, *Statistics Toolbox User's Guide*, The MathWorks, 2004.

[9]L.K.Saul and D.D. Lee, "Multiplicative Updates for Classification by Mixture Models", *NIPS 2001* Proceedings 2001.

[10]I.S.Dhillon and S.Sra, "Generalized Nonnegative Matrix Approximations with Bregman Divergences", *NIPS 2005* Proceedings, 2005.

[11]Y.Wang, Y.Jia, C.Hu and M.Turk, Fisher Non-Negative Matrix Factorization for Learning Local Features, *Proc. Asian Conf. on Comp. Vision*, 2004.

[12]E.G.Allen, M.R.Horvath, C.V.Kopek, B.T.Lamb, T.S.Whaples and M.W.Berry, Anomaly Detection Using Non-negative Matrix Factorization, *Proc. SIAM International Conference on Data Mining*, 2007.

[13]Seventh SIAM International Conference on Data Mining (SDM 2007), Text Mining Workshop's text mining competition website at http://www.cs.utk.edu/tmw07/.

[14]C.-W. Hsu, C.-C. Chang, and C.-J. Lin. "A practical guide to SVM classification." Technical report, Department of Computer Science and Information Technology, National Taiwan University, 2003.

[15]Duda, R., Hart, P., Stork, D., *Pattern Classification*, 2nd edR John Wiley & Sons, 2001.

[16]N. Christiani and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernelbased learning methods*, Cambridge University Press, 2000.

[17]C.J.C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, 2(2):955-974, 1998.

[18]Hettich, S. and Bay, S. D. (1999). The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science.

[19]Wolfe, S. (2007). "Wordplay: An Examination of Semantic Approaches to Classify Safety Reports", *AIAA Infotech@Aerospace*, AIAA Paper-2007-2821, Rohnert Park, CA, May 2007.