DIMACS Technical Report 2004-40
August 2004

Category-Based Feature Extraction in Supervised Categorization of Aviation Safety Report System Documents

by

Yangzhe Xiao,Haym Hirsh, Casimir A. Kulikowski, Michael L. Littman
Dept. of Computer Science
Rutgers University
New Brunswick, New Jersey 08854

Ilya Muchnik
DIMACS
Rutgers University
New Brunswick, New Jersey 08854

Sarah Zelikovitz
Computer Science Department
The College of Staten Island of CUNY
Staten Island, NY 10314

DIMACS is a collaborative project of Rutgers University, Princeton University, AT&T Labs–Research, Bell Labs, NEC Laboratories America and Telcordia Technologies, as well as affiliate members Avaya Labs, HP Labs, IBM Research and Microsoft Research. DIMACS was founded as an NSF Science and Technology Center.

ABSTRACT

In this study, we introduce novel feature extraction approaches for text categorization, which are based on characteristic representations of categories. A new category-based feature is derived from the relationship between a document and such a representation. A document is then projected onto the category-based coordinates in the new feature space. We evaluate two different approaches for extracting category-based features. One is a category-specific weighting, in which a representation of a category is a vector in the original space. The elements of the vector are the relative discriminating powers of all terms with respect to the category. The new feature of a document based on this category is the inner product between the representative vector and the document. The other is classifier-based, where every category-based coordinate is defined by a learned classifier. A new feature of a document on one coordinate is the value assigned by the discriminate function associated with the coordinate. We evaluate our new feature extraction methods for the Aviation Safety Report System documents using Support Vector Machines. The new category-based feature extraction methods give comparable results to the best ones obtained using feature selection with Chi-square based term ranking.

1. Introduction

Text categorization (or classification) is the assignment of natural language texts to one or more predefined categories (or classes) based on their content. It is an important component in many information organization and management tasks. Many statistical classification and machine learning techniques have been used in text categorization, including multivariate regression models (Yang and Chute, 1994), nearest neighbor classifiers (Yang, 1994), Naïve Bayes probabilistic classifiers (Lewis, 1992), decision trees (Lewis and Ringuette, 1994), neural networks (Schütze et al., 1995) and support vector machines (Joachims, 1998). These techniques generally infer class identity of unclassified documents by learning characteristics of categories from preclassified documents.

The art of text classification starts with the design of an appropriate representation method for text documents. A document is usually represented as a feature vector, as is typically done in the popular vector representation for information retrieval (Salton & McGill, 1983). For text classification that usually involves huge numbers of terms, the dimensionality becomes too big if one takes every unique term in documents as a feature. Dimension reduction is required for efficiency and defying the curse of dimensionality. Various techniques are proposed to address the tasks involving many irrelevant and redundant variables and often comparably few training examples. These techniques can be roughly grouped into two types: feature selection and feature transformation/extraction. The main objective is to improve prediction performance. The potential benefits also include reducing storage requirements, reducing training time, and facilitating data understanding.

Feature selection is a process that can choose a subset from an original term set according to some criteria. The idea is to choose terms that are strongly relevant to the classification performance. A simple technique is to rank terms according to some relevance measurements, e.g. Chi-square (Schütze et al., 1995; Yang and Pedersen, 1997; Yang and Liu, 1999), mutual information (Dumais et al., 1998; Yang and Pedersen, 1997), information gain (Yang and Pedersen, 1997; Yang and Liu, 1999). These measurements try to capture the intuition that the best terms for a category are the ones distributed most differently in different categories. Term ranking is known for its simplicity, scalability, and good empirical success. But it measures variables individually without considering the interaction between terms and predictors. The wrapper approach (John et al., 1994) avoids this problem by evaluating the prediction performance of a predictor with different subset of variables. The term set that results in the best effectiveness is chosen. This approach has the advantage of being tuned to the predictor being used, but it requires massive amount of computation.

Alternatively, feature extraction can be a process that constructs a set of new features from an original term set through some functional mapping. One technique is using clusters of words as features (Lewis, 1992; Baker and McCallum, 1998). Term clustering tries to group words with semantic relatedness, which targets to the problems of polysemy, homonymy and synonymy of words. Instead of clustering words, latent semantic indexing (LSI) (Deerwester et al., 1990) represents text documents in a low-dimensional "topic" space: the documents are characterized by some underlying (latent, hidden) concepts referred to by words. The principal component analysis (PCA) (Jolliffe, 1986) technique projects documents onto a space defined by the eigenvectors of the covariance matrix of variables. Such projection captures as much the variation of the data as possible.

In this paper, we introduce novel feature extraction approaches based on characteristic representations of categories. This idea of using category-based representation has been used in Kulikowski's paper (1971), in which each class was associated with a subspace and data vectors were discriminated by their distances from the class-specific subspaces. While in our approaches all classes share the same feature space and each coordinate in the feature space characteristically describes a category. New features of a document in the space are extracted based on the relationships between the document and those characteristic representations of categories. Such feature extraction can be geometrically interpreted as projecting documents onto a set of category-based coordinates in the new feature space. We evaluate two different approaches based on this idea: category-specific weighting and classifier-based features. We compare our new feature extraction methods with feature selection using Chi-square based term ranking in the classification of the Aviation Safety Report System (ASRS) documents. We use Support Vector Machines (SVMs) as our classification method. In section 2, we introduce the feature selection and extraction approaches used in this study. Section 3 has the experiments and results. Section 4 has the discussion.

2.Document representation

Suppose D is a document set and C is a category set. A text classification problem can be formalized as finding a predictor function $f\colon\ D \rightarrow C$. The set D can be randomly partitioned into two sets: training T_r and testing T_e. The training set is used to train the predictor, while the testing set is used for estimating the generalization error of the predictor.

2.1 Feature selection using term ranking

We use term ranking as a comparison with our new feature extraction methods for its simplicity. The criterion used to rank terms is the well-known Chi-square test. Yang and Pedersen (1997) compared five measurements (document frequency, information gain, mutual information, Chi-square test, and term strength) in text categorization using different learning algorithms, and found that the Chi-square test and information gain gave the best performance. We selected the Chi-square test as our measurement in ranking terms.

	c	Non-c
t	A1	A2
Non-t	A3	A4

Table 1. Two-way contingency table of a term t and a category c.

We can build a two-way contingency table (Table 1) of a term t and a category c using documents in T_r, where A1 is the number of documents in c that have t, A2 is the number of documents that are not in c and have t, A3 is the number of documents in c that do not have t, and A4 is the number of documents that are not in c and do not have t. The Chi-square test measurement of term t with respect to category c is defined as

$$\chi^2(t,c) = \frac{|T_r| \times (A1A4 - A2A3)^2}{(A1+A3)\times(A2+A4)\times(A1+A2)\times(A3+A4)},$$

where $|T_r|$ is the number of training documents.

From the formula, we can see $\chi^2(t,c)$ is close to zero if t and c are independent. If the distribution of t is uneven between c and non-c documents, t is informative in discriminating c from non-c and $\chi^2(t,c)$ has a bigger value. Therefore, $\chi^2(t,c)$ measures the relative discriminating power of t with respect to c. A term will have different χ^2 values with respect to different categories. The maximum value of a term among all categories is used as the ranking value of the term. Only a certain number of top ranking terms are chosen to form the feature set F. We will refer to this feature selection method as "ChiFS" in the following sections.

A document can be represented as a feature vector using Term Frequency Inverse Document Frequency (TFIDF) (Salton and Buckley, 1988). The value of each feature of a document vector is calculated as

$$tfidf(t_k, d_j) = tf(t_k, d_j) * log(|T_r| / df(t_k)),$$

where $tf(t_k, d_j)$ denotes the number of times that term t_k occurs in document d_j in T_r, and $|T_r|$ is the size of training set. $df(t_k)$ is the document frequency of term t_k, i.e. the number of documents in T_r that have term t_k (Salton and Buckley, 1988). This calculation combines the two arguments that (i) the more often a term occurs in a document, the more it is representative of its content, and (ii) the more documents a term occurs in, the less important it is. The document vectors can be normalized to have unit length to mitigate the influence of variable length of documents. The normalized value w_{kj} is calculated as

$$w_{kj} = \frac{tfidf(t_k,d_j)}{\sqrt{\sum_{k=1}^{|F|}(tfidf(t_k,d_j)^2}},$$

where $|F|$ is the cardinality of the feature set F.

2.2 Feature extraction using category-specific weighting

In the TFIDF formula, *tf*(t_k,d_j) varies with term t_k and document d_j. The second part of the formula is constant for t_k, which can be considered as a weight assigned to each term. This weight-assigning scheme is not based on statistical relevance of a term in discriminating categories. Instead, it is based on the intuition that the more frequent a term is, the less discriminating it is. But we can see that such weighting strategy will fail in some situations. For example, a term may occur frequently in a category, but seldom in other categories. This term will be ranked high using $\chi^2(t,c)$-based ranking. However, TFIDF will assign a small weight to the term if the documents of that category contribute significantly to the size of training set. Hence, the weighting strategy in TFIDF is neither robust nor category-specific.

The weighting strategy that we propose here uses robust relevance measurements for terms with respect to individual categories. Such a measurement assigns weight $w(t_j,c_i)$ to a term t_j for a individual category c_i. A $|C| \times |F|$ weight matrix $\mathbf{W}$ is formed with $w(t_j,c_i)$ as the ith row and jth column element. Each row vector of $\mathbf{W}$ represents the relative weights of all features in F for the corresponding category.

To extract new features from a document vector $\mathbf{x}$ in the original space defined by the feature set F, we multiply $\mathbf{W}$ by $\mathbf{x}$. The result of such multiplication is a new vector in which each element is the inner product of a row vector of $\mathbf{W}$ and $\mathbf{x}$. Since each row vector in $\mathbf{W}$ is the category-specific weights of all features in the original space, the inner product is the weighted sum of all features for the document. The dimensionality of the new vector is equal to $|C|$, i.e. the number of rows of $\mathbf{W}$. We can think of each row vector of $\mathbf{W}$ as a concept or characteristic of the corresponding category. Then, each dimension in the new space captures the relationships between documents and one category. By choosing robust measurements for $w(t_j,c_i)$, this weighting scheme is not only robust but also category-specific.

One interesting direction in kernel based-learning algorithms is the modification of the kernel function with extra domain knowledge. It is known that an appropriate kernel can make linearly non-separable data become separable. The modification of kernel functions aims at improving prediction accuracy for domain-specific problems. For example, in natural images, correlations over short distances are much more reliable as features than long-range correlations, so there is an advantage for using the constructed kernel corresponding to a dot product in a space that is spanned mainly by local correlations between pixels (Scholkopf et al., 1998). In text classification, using common kernels often results in the semantic relations between terms not being taken into account. Documents on the same topic using different terms are possibly mapped to very distant regions in feature space. Encoding a semantic network into kernel function maps implicitly documents into a “semantic space” (Siolas and D'Alche-buc, 2000; Christianini et al., 2001).

Our category-specific weighting method can be viewed as a modification of a kernel with the weight matrix $\mathbf{W}$. For example, if using a Gaussian kernel, our weighting method is equivalent to

$$\mathrm{K}(\mathrm{x}_i,\ \mathrm{x}_j)=\exp(-||\mathbf{W}\mathrm{x}_i\ \text{-}\ \mathbf{W}\mathrm{x}_j||^2/\ 2\sigma^2).$$

Therefore, the category-specific weighting method can be viewed as implicitly mapping documents into a "weighted space".

There are many possible statistical measurements for $w(t_j,c_i)$. We use two different measurements in this feature extraction scheme. The first one is the $\chi^2(t,c)$ that has been introduced in section 2.1. It gives robust statistical measurement for the relative discriminating power of a term. We will call category-specific weighting with $\chi^2(t,c)$ as "CFE". From the two-way contingency table 1 and the formula, we can see $\chi^2(t,c)$ gives equal emphasis to frequent and rare terms as long as the distributions of the terms are uneven among different categories. Also $\chi^2(t,c)$ is a symmetric measurement, which may cause problems for data that have only two categories. In this case, $\chi^2(t,\bullet)$ will be the same for the two categories.

The second one is $E(t,c)$ designed for weighting term t with respect to category c, which is defined as

$$E(t,c) = \tfrac{A1}{A1+A3} \times (1 + p\log_2 p + q\log_2 q),$$

where

$$p = \frac{\frac{A1}{A1+A3}}{\frac{A1}{A1+A3}+\frac{A2}{A2+A4}}, \qquad q = 1 - p,$$

and A1, A2, A3 and A4 have the same meanings as those in table 1. The parenthesized part in $E(t,c)$ ranges from 0 to 1, measuring the unbalanced distribution of t in c and non-c. It has value 1 if t only appears exclusively in either c or non-c documents. This part, just like $\chi^2(t,c)$, gives big weight if t is distributed unevenly among categories. It is possible that t is a rare term in c and still scores high from this part. The first part in the formula calculates the proportion of category c that has t, which favors frequent terms over rare ones. It also makes $E(t,c)$ asymmetric. Therefore, $E(t,c)$ emphasizes both an unbalanced distribution across categories and a high intra-category document frequency of a term. We will refer to category-specific weighting with $E(t,c)$ as "EFE".

2.3 Feature extraction using classifiers

In supervised classification, a binary classifier is trained to separate positive and negative samples. The decision of a classifier on a document is based on the relationships between the document and the training samples involved. Therefore, the values assigned by such a discriminate function for the documents represent the relative "closeness" of the documents to the categories involved. These values can be used as extracted new features. Since all classifiers are category-based, so are the new features.

In our extraction method, a set of classifiers is first obtained using a training set. One type of classifiers is trained using one category as positive and all the other categories as negative, which is a one-vs.-others strategy. Another type of classifiers is trained with only two categories. We use both type of classifiers to capture the subtle relationships among categories. Given that the number of categories is $|C|$, the number of classifiers in the set is $|C| + |C| \times (|C|$ - 1) / 2, which is also equal to the dimensionality of the new space. Each document is classified using this set of classifiers. The values resulted from these classifiers are used as new features. We can see that each new feature represents the "opinion" of a specific classifier on the relationships between a document and the categories involved.

This idea is different from stack generalization (Wolpert, 1992), in which the results of the first-level classifiers are fed as input to a second-level predicator to generate output. This stacked structure of classifiers aims at correcting the biases of the constituent first-level classifiers. So for a multi-category problem, each category will have its own first-level classifiers and the second-level classifier is trained on different sets of input. In our method, we are interested in deriving category-based features from the classifiers obtained in the original space. The training and testing are carried out after every document is transformed into the new feature space. We will refer to this classifier-based feature extraction method as "VFE".

3. Results

The documents in this study are aviation safety reports from the Aviation Safety Reporting System (ASRS). The reports are submitted voluntarily by pilots, controllers, air carriers or other participants in the aviation systems. Each report has a "Narrative" section that describes anomalies and incidents. The "Narrative" sections of the reports were taken as our text documents.

We collected two different sets of reports. Periodically ASRS publishes a categorized report set in which each report has been screened by human experts to assure its relevance to some topics. The report set published in Dec 2002 has 28 topics (See Appendix Table 1) and 1380 reports. We refer to it as "L-set". We also downloaded from ASRS database more than 80,000 reports, which form our "U-set". These documents don't have the same category labels as those in the L-set. Each report in the U-set has a section called "Anomaly Descriptors". The entries in "Anomaly Descriptors" are not the same as the 28 topics in the L-set. We categorized documents in the U-set into 10 topics (Appendix Table 2) based on these entries. In both sets, a document can be multi-labeled. All documents underwent the preprocessing step: removing non-content bearing words (stop words) and stemming.

SVM-light software (Joachims, 1999) was used for training and testing. The SVMs method has showed very good performance in text classification and has advantages of robustness to overfitting and scalability up to considerable dimensionality. In addition, SVMs are well suited for both dense and sparse problems (Joachims, 1998). Our feature extraction methods result in dense document vectors. Hence, we chose to use SVMs as our learning algorithm.

Four performance measurements are used in the following results tables. 1) Precision (π) is the proportion of documents assigned to a category c that are really in the category. It can be viewed as the measurement of "degree of soundness". 2) Recall (ρ), also known as sensitivity, is the proportion of documents of c that are correctly assigned to the category. It measures how sensitive that a classifier is in picking out positive samples. 3) Specificity is the proportion of non-c documents that are classified as non-c. 4) F1 is defined as

$$F1 = \frac{2\pi\rho}{\pi+\rho},$$

which depends on both precision and recall. In the following tables, precision, recall and specificity are micro-averages[1].

For CFE, EFE and VFE, the documents were represented using normalized TFIDF with all the terms left after the preprocessing step. Then the document vectors were transformed via the respective feature extraction methods. For ChiFS, a series of training and testing with different numbers of top ranking terms were carried out to find the suitable size of feature set.

	Precision	Recall	Specificity	F1
ChiFS	0.65	0.55	0.99	0.59
CFE	0.74	0.59	0.99	0.66
EFE	0.72	0.57	0.99	0.64
VFE	0.77	0.45	0.99	0.57

Table 2.Classification results of the four methods in the L-set.

The L-set has 28 categories and each category has approximately 50 reports. It has about 7000 unique terms after preprocessing. In ChiFS, we found that the top 4,000 and 6,000 ranked terms gave the best results (i.e. highest F1 values). We chose 4,000 as the size of the selected feature set in ChiFS. The 5 fold cross validation classification results of the four methods are shown in Table 2. The precision, recall, and specificity are micro-averages over the 28 categories. All the methods have very good results in specificity, which are close to 1. But sensitivity (i.e. recall) is around 0.5. CFE and EFE are a little better than ChiFS and VFE in terms of F1 values.

[1]Micro-average of precision over category set C is calculated as $\pi = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i+FP_i)}$, where TP_i and FP_i stand for true positive and false positive w.r.t. category i. Micro-averages of recall and specificity are calculated with their corresponding formulas in the same way.

The individual results of each method in the 28 categories are listed in Appendix tables 3-6. In general, for some categories like “Animal”, “Chklist”, “Fuel”, “Mechanic”, “Parachute”, “Ped” and “Waketurb”, it is easy to get good classifiers with F1 values > 0.8 in any of the four methods, while the four methods show poor performance in some categories with F1 values around 0.4, such as “Rec_ops”, “Upsets”, “Rwy_incur”, “Altdev” and “Penetrat”. The four methods show very different results in some categories. For example, VFE does very poor in category “Helo”, while the other three have much better F values.

The U-set has 10 categories. The number of documents in these 10 categories ranges from 2416 to 19,900. We found that in ChiFS the feature set size of 2,000 to 6,000 had very similar results. It was appealing to choose 2,000 as the feature set size for efficiency. The 3 fold cross validation results are listed in table 3. The precision, recall, and specificity are micro-averages over the 10 categories. The four methods have very similar results in terms of F1 values.

	Precision	Recall	Specificity	F1
ChiFS	0.78	0.57	0.97	0.66
CFE	0.81	0.59	0.97	0.68
EFE	0.83	0.56	0.98	0.67
VFE	0.71	0.62	0.95	0.66

Table 3.Classification results of the four methods in the U-set.

The individual results of each method in the 10 categories are listed in Appendix tables 7-10. For some categories, such as 1, 2, 6 and 7, we got better classifiers using all the four methods. We observed that if a category had a poor classifier under one method, it was very likely to see poor classification performance under the other three methods. An extreme example is category 3, where all the four methods failed to identify any positive samples and all had 0 in recall. The performance of the four methods seems category-related. None of the four methods is consistently better than the rest in all categories of both the L-set and U-set.

To see how the category-specific weighting method using the Chi-square test affects the classification performance, we calculate some statistics (Table 4) on all the documents in L-set. Column 2 has top 5 terms ranked by the Chi-square test for every category. Since the original documents have a lot abbreviations and stemming was used, the terms listed in column 2 may not be words. But we still can see that some terms are quite relevant to their categories. For example, category “Animal” has “bird”, “flock”, “strike”, and “deer” in the top 5. The ratios between the weights of the top 50 terms and all terms are listed in column 3, which has a mean of 0.123. This number is fairly large given the total number of terms is above 7,000. The ratios between the weights of the top 50 terms and the bottom 50 terms are listed in column 4, which has a mean of 968. The two means of column 3 and 4 indicate that the weighted sums (new features) depend highly on the top ranking several hundred terms.

Categories	Top 5 terms	Top 50/Total	Top50/Bott50
acr_fatg	fatigu slept overnight snowbal ordinari	0.096	288
altdev	lda patti wesen effi tusti	0.123	649
animal	bird flock strike deer ingest	0.148	2437
awos	awo aso decent weathervan nuclear	0.088	485
cabin	gallei aisl attend beverag lav	0.113	1269
cftt	terrain egpw gpw elm toga	0.123	845
chklist	qrh eica driven overheat memori	0.112	634
com_fatigue	fatigu awak rerout acton evolv	0.083	150
crm	crm priorit norm inclin subtl	0.088	184
ctlr	sector loa vxv brite decombin	0.151	2239
fuel	fuel tank gallon starvat gph	0.108	612
ga_train	student instructor client courteou cfi	0.108	287
gps	gp map gso kln avn	0.100	450
helo	heli helicopt hover label institut	0.109	566
icing	ic rime boot accumul moistur	0.102	314
lahso	lahso lnd short hnl autobak	0.158	1185
mechanic	instal svcing technician reinspect nut	0.140	1799
non_twr	ctaf pattern announc gentleman hei	0.108	652
parachute	skydiv jumper jump parachut olympia	0.172	1603
ped	devic cell electron laptop ped	0.124	1603
penetrat	tfr text dc pilotag maryland	0.099	752
plt_ctlr	phraseolog miscom nonstandard clarif stopway	0.152	1296
rec_ops	home built plant privat popul highwai	0.101	1040
rwy_incur	txwy centerlin logo nighttim dimens	0.145	1114
tcas2	ra tcasii ta tcasi intrud	0.189	1705
upsets	trim yaw damper stiff recess	0.091	649
waketurb	wake turb offset trail counteract	0.182	1536
wx	lightn windshear sever chop turb	0.123	755

Table 4. Statistics of the Chi-Square based weighting in the 28 categories of the L-set.

4. Discussion

In this study, we introduce two novel feature extraction approaches based on category-related characteristics: category-specific weighting and classifier-based features. The results of our new approaches are comparable to the best results obtained with feature selection with Chi-square based term ranking in the ASRS domain using SVMs. Although the L-set is human-expert labeled, the F1 values of the U-set data are better than those of the L-set. The possible explanation is that the size of the training data is much larger in the U-set (Total 80,000 documents vs. 1380 documents in the L-set), which results in a better representation of the distribution of the data.

Also the document vectors generated with our feature extraction methods are dense, which may also cause problems to some learning algorithms that require sparse vectors. Using the category-specific weighting method the dimension of the new space is the same as the number of categories. This can be a dramatic dimensionality reduction, like the reduction in the U-set data from more than 45,000 to 10. But too few categories may negatively impact classification accuracy. For an extreme case in which only two categories exist, it is very likely that some useful information in the original high dimensional space would get lost after being transformed into the two-dimensional space. But it is difficult to know what is the maximum reduction before incurring unacceptable deterioration of accuracy.

To the best of our knowledge, there is not any publication on text classification using SVMs in this domain. That was one of the reasons we chose this domain, hoping that our study could help in this practical problem. Our results show that some categories in the ASRS data have very good classifiers, which can be a great help in automatic assignment for these categories. Even for the data in the U-set to which we assigned the labels, good classifiers are available for some categories.

In the VEF method, it is computationally expensive to obtain a set of classifiers from the original space using SVMs. The high computation cost for the VFE method restricts its practical use if it does not bring about a large boost in accuracy. It is reasonable to use some computationally cheap algorithms in preparing classifiers for feature extraction. In our experiment, we collected a set of classifiers from one-vs.-others and every pair of categories in order to capture as many relationships among categories as possible. It is possible that a selective subset of these classifiers may be good enough, which would lower the cost.

The current results demonstrate that category-based feature extraction is an applicable and valuable idea. Our future work involves evaluating these new approaches on data from different domains with different learning algorithms. An important derivative we will study is cluster-based feature extraction. Since clustering is an unsupervised learning method, we can use both labeled and unlabeled documents to get clusters. Therefore, the cluster-based feature extraction will exploit the natural patterns of all the data. We expect incorporation of unlabeled data in feature extraction will improve accuracy.

Appendix

Categories	Full Topic Names
acr_ftag	Air Carrier (FAR 121) Flight Crew Fatigue Reports
altdev	Altitude Deviations
animal	Bird or Animal Strike Reports
awos	Automated Weather Systems
cabin	Cabin Attendant Reports
cftt	Controlled Flight Toward Terrain
chklist	Checklist Incidents
com_fatigue	Commuter and Corporate Flight Crew Fatigue Reports
crm	Cockpit Resource Management (CRM) Issues
ctlr	Controller Reports
fuel	Fuel Management Issues
ga_train	General Aviation Flight Training Incidents
gps	Global Positioning System (GPS) Reports
helo	Rotary Wing Aircraft Flight Crew Reports
icing	Commuter and GA Icing Incidents
lasho	Land and Hold Short Operations
mechanic	Maintenance Reports
non_twr	Non-Tower Airport Incidents
parachute	Parachutist/Aircraft Conflicts
ped	Passenger Electronic Devices
penetrat	TFR/ADIZ Airspace Incidents
plt_ctlr	Pilot/Controller Communications
rec_ops	Recreational Aviation Incidents
rwy_incur	Runway Incursions
tcas2	TCAS II Incidents
upsets	Multi-Engine Turbojet Aircraft Upsets Incidents
waketurb	Wake Turbulence Encounters
wx	Inflight Weather Encounters

Table 1. Names of the 28 categories in the L-set.

Categories	Description
0	No description and ASP Anomaly
1	Acft Equip Anomaly
2	Non-Adherence Anomaly
3	Non-Adherence Other Anomaly
4	Other Anomaly
5	Other Spatial Dev. Anomaly
6	Alt Dev Anomaly
7	Conflict Anomaly
8	Incursion Anomaly
9	In-Flight Anomaly

Table 2. Names of the 10 categories in the U-set.

Categories	Precision	Recall	Specificity	F1
acr_fatg	0.652	0.438	0.998	0.524
altdev	0.271	0.347	0.991	0.304
animal	0.960	0.727	0.999	0.827
awos	0.859	0.630	0.995	0.727
cabin	0.764	0.560	0.990	0.646
cftt	0.586	0.583	0.995	0.584
chklist	0.867	0.755	0.992	0.807
com_fatigue	0.471	0.340	0.995	0.395
crm	0.886	0.486	0.999	0.628
ctlr	0.475	0.310	0.997	0.375
fuel	0.733	0.950	0.993	0.828
ga_train	0.615	0.422	0.995	0.500
gps	0.768	0.580	0.999	0.661
helo	0.386	0.360	0.999	0.373
icing	0.848	0.667	0.998	0.746
lahso	0.886	0.620	0.999	0.730
mechanic	0.845	0.833	0.998	0.839
non_twr	0.639	0.449	0.992	0.527
parachute	0.976	0.816	1.000	0.889
ped	0.947	0.757	0.999	0.841
penetrat	0.365	0.393	0.985	0.378
plt_ctlr	0.405	0.453	0.981	0.428
rec_ops	0.265	0.116	0.995	0.161
rwy_incur	0.306	0.411	0.970	0.351
tcas2	0.623	0.732	0.993	0.673
upsets	0.435	0.200	0.995	0.274
waketurb	0.947	0.783	0.999	0.857
wx	0.798	0.628	0.994	0.703
Micro-Avg	0.647	0.547	0.994	0.593

Table 3. Classification results of ChiFS in the L-set.

Categories	Precision	Recall	Specificity	F1
acr_fatg	0.741	0.417	0.995	0.533
altdev	0.400	0.103	0.995	0.163
animal	0.918	0.900	0.997	0.909
awos	0.850	0.680	0.995	0.756
cabin	0.875	0.700	0.996	0.778
cftt	0.647	0.660	0.986	0.653
chklist	0.809	0.776	0.993	0.792
com_fatigue	0.462	0.300	0.989	0.364
crm	0.941	0.667	0.998	0.780
ctlr	0.655	0.380	0.992	0.481
fuel	0.784	0.851	0.992	0.816
ga_train	0.649	0.490	0.990	0.558
gps	0.816	0.620	0.995	0.705
helo	0.453	0.480	0.978	0.466
icing	0.884	0.809	0.996	0.844
lahso	0.878	0.720	0.996	0.791
mechanic	0.905	0.776	0.997	0.835
non_twr	0.694	0.510	0.992	0.588
parachute	0.956	0.878	0.998	0.915
ped	0.867	0.780	0.995	0.821
penetrat	0.514	0.380	0.986	0.437
plt_ctlr	0.643	0.360	0.992	0.462
rec_ops	0.375	0.184	0.989	0.247
rwy_incur	0.476	0.400	0.983	0.435
tcas2	0.769	0.784	0.991	0.777
upsets	0.391	0.180	0.989	0.247
waketurb	0.898	0.898	0.996	0.898
wx	0.821	0.667	0.995	0.736
Micro-Avg	0.741	0.589	0.992	0.656

Table 4. Classification results of CFE in the L-set.

Categories	Precision	Recall	Specificity	F1
acr_fatg	0.803	0.417	0.997	0.549
altdev	0.200	0.082	0.988	0.116
animal	0.973	0.880	0.999	0.924
awos	0.737	0.680	0.991	0.707
cabin	0.690	0.700	0.988	0.695
cftt	0.767	0.540	0.994	0.634
chklist	0.738	0.673	0.991	0.704
com_fatigue	0.543	0.184	0.995	0.274
crm	0.889	0.688	0.997	0.775
ctlr	0.733	0.360	0.995	0.483
fuel	0.840	0.894	0.994	0.866
ga_train	0.619	0.388	0.991	0.477
gps	0.905	0.540	0.999	0.677
helo	0.667	0.480	0.991	0.558
icing	0.915	0.787	0.998	0.847
lahso	0.848	0.780	0.995	0.813
mechanic	0.847	0.892	0.995	0.869
non_twr	0.607	0.510	0.987	0.554
parachute	0.990	0.837	1.000	0.907
ped	0.892	0.800	0.996	0.843
penetrat	0.438	0.400	0.980	0.418
plt_ctlr	0.417	0.380	0.979	0.397
rec_ops	0.435	0.224	0.990	0.296
rwy_incur	0.340	0.420	0.968	0.376
tcas2	0.769	0.745	0.991	0.757
upsets	0.571	0.200	0.994	0.296
waketurb	0.966	0.910	0.999	0.937
wx	0.750	0.708	0.991	0.729
Micro-Avg	0.724	0.574	0.992	0.641

Table 5. Classification results of EFE in the L-set.

Categories	Precision	Recall	Specificity	F1
acr_fatg	0.830	0.223	0.998	0.352
altdev	0.256	0.212	0.977	0.232
animal	0.931	0.812	0.993	0.867
awos	0.899	0.364	0.999	0.518
cabin	0.727	0.692	0.990	0.709
cftt	0.778	0.273	0.997	0.404
chklist	0.692	0.886	0.985	0.777
com_fatigue	0.274	0.101	0.980	0.148
crm	0.732	0.252	0.997	0.375
ctlr	0.853	0.121	0.999	0.212
fuel	0.854	0.976	0.994	0.911
ga_train	0.788	0.201	0.998	0.320
gps	0.784	0.463	0.996	0.582
helo	0.687	0.112	0.998	0.193
icing	0.857	0.623	0.996	0.722
lahso	0.889	0.727	0.997	0.800
mechanic	0.954	0.823	0.999	0.884
non_twr	0.865	0.556	0.997	0.677
parachute	0.943	0.912	0.998	0.927
ped	0.901	0.546	0.998	0.680
penetrat	0.723	0.182	0.997	0.291
plt_ctlr	0.563	0.112	0.997	0.187
rec_ops	0.893	0.152	1.000	0.260
rwy_incur	0.287	0.125	0.988	0.174
tcas2	0.887	0.546	0.997	0.676
upsets	0.520	0.304	0.989	0.383
waketurb	0.973	0.800	0.999	0.878
wx	0.859	0.375	0.998	0.522
Micro-Avg	0.774	0.448	0.994	0.568

Table 6. Classification results of VFE in the L-set.

Categories	Precision	Recall	Specificity	F1
0	0.750	0.225	0.991	0.346
1	0.888	0.722	0.970	0.796
2	0.765	0.688	0.704	0.725
3	NAN	0.000	1.000	NAN
4	0.743	0.310	0.994	0.437
5	0.732	0.342	0.980	0.466
6	0.754	0.635	0.959	0.689
7	0.764	0.627	0.936	0.688
8	0.702	0.350	0.989	0.467
9	0.744	0.376	0.991	0.499
Micro-Avg	0.779	0.573	0.967	0.660

Table 7. Classification results of ChiFS in the U-set.

Categories	Precision	Recall	Specificity	F1
0	0.772	0.193	0.993	0.309
1	0.889	0.711	0.971	0.790
2	0.785	0.739	0.717	0.761
3	NAN	0.000	1.000	NAN
4	0.779	0.296	0.995	0.429
5	0.829	0.320	0.989	0.462
6	0.808	0.662	0.969	0.728
7	0.816	0.612	0.954	0.700
8	0.741	0.363	0.991	0.487
9	0.816	0.385	0.994	0.523
Micro-Avg	0.810	0.585	0.972	0.680

Table 8. Classification results of CFE in the U-set.

Categories	Precision	Recall	Specificity	F1
0	0.786	0.186	0.994	0.301
1	0.894	0.702	0.973	0.786
2	0.809	0.713	0.764	0.758
3	NAN	0.000	1.000	NAN
4	0.797	0.274	0.996	0.408
5	0.856	0.303	0.992	0.447
6	0.829	0.636	0.974	0.720
7	0.824	0.585	0.958	0.684
8	0.768	0.327	0.993	0.458
9	0.825	0.369	0.995	0.510
Micro-Avg	0.828	0.564	0.976	0.671

Table 9. Classification results of EFE in the U-set.

Categories	Precision	Recall	Specificity	F1
1	0.822	0.743	0.947	0.781
2	0.763	0.677	0.705	0.717
3	NAN	0.000	1.000	NAN
4	0.622	0.466	0.985	0.533
5	0.653	0.443	0.962	0.528
6	0.682	0.723	0.933	0.702
7	0.713	0.644	0.914	0.677
8	0.601	0.484	0.977	0.536
9	0.623	0.504	0.979	0.557
Micro-Avg	0.714	0.615	0.949	0.661

Table 10. Classification results of VFE in the U-set.

References

Baker, L.D. and Mccallum, A.K. Distributional clustering of words for text classification. In *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, 96–103, 1998.

Christianini, N., Shawe-Taylor, J. and Lodhi, H. Latent semantic kernels. *Proc. of the 18th International Conference on Machine Learning*, 66-73. 2001.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R. 1990. Indexing by latent semantic indexing. *J. Amer. Soc. Inform. Sci.* 41, 6, 391–407.

Dumais, S.T., Platt, J., Heckerman, D. and Sahami, M. Inductive learning algorithms and representations for text categorization. In *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, 148–155, 1998.

Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings 10th European Conference on Machine Learning (ECML)*, 137-142, 1998.

Joachims, T. Making large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press,* 1999.

John, G.H., Kohavi, R. and Pfleger, K. Irrelevant features and the subset selection problem. In *Proceedings of ICML-94, 11th International Conference on Machine Learning*, 121–129. 1994.

Jolliffe, I.T. Principal Component Analysis. Springer Series in Statistics. 1986.

Kulikowski, C.A. Discriminatory Dimensionality reduction. *IEEE Transactions on Information Theory*, 498-499, 1971.

Lewis, D.D. An evaluation of phrasal and clustered representations on a text categorization task. In *SIGIR'92: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 37-50, 1992.

Lewis, D.D. and Ringuette, M. A comparison of two learning algorithms for text categorization. *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval (Las Vegas, NV.)*, 81-93, 1994.

Salton, G. and McGill, M. Introduction to Modern Information Retrieval. McGraw Hill, 1983.

Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513-523, 1988.

Scholkopf, B., Simard, P., Smola, A. and Vapnik, V. Prior knowledge in support vector kernels. *Advances in Neural Inf. Proc. Systems, volume* 10. 640-646, 1998

Schütze, H., Hull, D. and Pedersen, J.O. A comparison of classifiers and document representations for the routing problem. In *SIGIR '95: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 229-237, 1995.

Siolas, G. and D'Alche-buc, F. Support vector machines based on a semantic kernel for text categorization. *In Proceedings of the International Joint Conference on Neural Networks, IJCNN, Como,. IEEE.* Vol.5, 205-209, 2000.

Wolpert, D.H. Stacked generalization. *Neural Networks,* 5(2), 241-259, 1992.

Yang, Y. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. *SIGIR '94: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 13-22, 1994.

Yang, Y. and Chute, C.G. An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems,* 12(3), 252-277, 1994.

Yang, Y. and Liu, X. A re-examination of text categorization methods. In *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval* (Berkeley, CA), 42–49, 1999.

Yang, Y. and Pedersen, J.O. A comparative study on feature selection in text categorization. *In Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97),* 412-420, 1997.