Advanced Text Mining Algorithms for Aerospace Anomaly Identification

Z. Bluvband & S. Porotsky
A.L.D, Tel-Aviv, Israel

ABSTRACT: Article describes the advanced text categorization procedure developed and successfully used in aerospace industry, especially for safety assessment, analysis and improvement. The purpose is the computerized analysis and interpretation of human reported free-text aviation safety records, in order to automatically "read", discover and treat anomalies occurred in the field. The methodology and algorithms were verified on actual, significant and appropriate ASRS (Aviation Safety Reporting System) data base (http://asrs.arc.nasa.gov/index.html) as well as other similar data bases containing millions of unprocessed safety and reliability reports. One of the most important applications and goals of the reasearch is to assign new incoming safety event reports to one or more from the several of predefined categories on the basis of their textual content.

Optimal categorization functions can be constructed from labeled training examples (i.e., after human expertise) by means of supervised learning algorithm and cross-validation. Numerous methods for text categorization have been developed lately: Neural Networks, Naive Bayes, AdaBoost, Linear Discriminant Analysis, Logistic Regression, Support Vector Machines (SVM), etc. SVM has become a popular learning algorithm, used in particular for large, high-dimensional classification problems; it has been shown to give most accurate classification results in a variety of applications. However the Direct application of these methods to Aerospace Anomaly Discovery is restricted for the following reasons:

a) fully automatic procedure can support only middle values of Recall and Precision (50-75 %);

b) lack of stability of the reports statistical parameters - i.e. the frequency of words in a report has been changing on a "year to year" basis.

To support high values of output criteria (e.g., both Recall and Precision have to be simultaneously more than 90-95 %) and non-stability of the report statistics, the mixed, partially automated approach was proposed for the selection of most of anomalies automatically, by means of text categorization algorithm, with ocasional usage of human expertise. Numerical example, based on ASRS On-Line Data Base, is considered.

1 INTRODUCTION

One of the most important activities during the safety-sensitive system's life cycle (design, development, test, production and operation) is continuous safety and reliability measurement and tracking, risks assessment for safety improvement and reliability growth. For this purpose all the events: incidents, accidents and failures (occurred or almost occurred) should be reported, recorded, retrieved, classified and analyzed. Typically these reports are human-written records, usually just free text written by professional people. Text mining is one of the most important tasks in such a business, and text categorization (classification) is a fundamental task in the text mining, in theory and in practice. Text categorization is the process of grouping written reported documents into different categories or classes. With the amount of online information growing rapidly, the need for reliable automatic text categorization has increased. Since a safety report as a text document often belongs to multiple categories, text categorization is generally defined as a methodology and an algorithm (classifier) for assigning one or more predefined category labels to certain data sample. The usual approach to solve this problem is based on the "supervised learning". It uses mathematical model "to learn" the relationship between a set of data and some known field category.

One of the most widely applied learning algorithms for text categorization is the relevance feedback method (Rocchio 1971, Joachims 1997) developed for information retrieval. This method is quite easy to implement, and is also quite efficient, since learning a classifier basically comes down to averaging weights. The classifier built by the Rocchio method is linear and, as all linear classifiers, has the disadvantage that it divides the space of documents linearly.

A Neural Network text classifier (Wiener et al. 1995) uses the idea of a network of units, where the input units represent terms, the output units represent the categories of interest, and the weights on the edges connecting units represent dependence relations. There are also other supervised learning algorithms which are used for text categorization like Logistic Regression, Naive Bayes, AdaBoost, Linear Discriminant Analysis, etc. (Dumais et al. 1998 , Lewis 1998, Srivastava et al. 2006).

Support Vector Machines (SVM) has become a popular learning algorithm, in particular for large, high-dimensional classification problems (Scholkopf & Smola 2001). SVM has been shown to give most accurate classification results in a variety of applications (Dumais et al. 1998, Srivastava et al. 2006). In SVM classification, the optimal separating function comes down to a linear combination of kernels on the training data with training feature vectors **X** and corresponding labels Y.

Usually, the performance of a classifier is measured in terms of accuracy based on the comparison of the classifiers prediction of the true class. But in some cases this is not enough because it doesn't give sufficient information. For example, for unbalanced data sets ("unbalanced" or "imbalanced" means data set which have much-much more – more than 97% – negative than positive examples) the optimal classifier may be by default a negative classifier. Prediction accuracy for Text Mining tasks with unbalanced dataset is usually estimated by a combination of two metrics – recall and precision:

$$recall = \frac{TP}{TP + FN}$$

$$precision = \frac{TP}{TP + FP}$$

Where:

FP (False Positives) is the number of negative examples - incorrectly classified as positive; it is amount of Type_1 Errors - to include "garbage" (take in a case as a positive instance when it is not).

FN (False Negatives) is the number of positive examples incorrectly classified as negative; it is amount of Type_2 Errors, i.e. errors of "loss" of really positive instances.

TP (True Positives) is the number of positive examples correctly classified.

Several SVM-based techniques may be used for imbalanced data sets categorization (Chawla et al. 2002, Akbani et al. 2004, Imam et al. 2006, Wu & Chang 2005). Nevertheless even these different and numerous tools sometimes don't allow receiving concurrently high values of both Recall and Precision. According to different articles, typical values for the Recall and Precision for the high-imbalanced data sets (amount of positive samples is less than 3 %) are not more than Break-Even Point when both of the metrics are around 0.5…0.7. Even for low-imbalanced data sets (amount of positive samples is 3…10 %) their typical values are not more than Break-Even Point of 0.7…0.8. The achieved results for ASRS On-Line Data Base Report's Categorization (Srivastava et al. 2006) are very similar – for different anomalies'– the values of Break-Even Point vary from 0.5 to 0.75. Unfortunately this is not enough because in many situations it is necessary to provide jointly high values (0.9…0.95) for both Recall and Precision. So the major common limitation of these prior art approaches is as following: their inability to support the high values of both Recall and Precision at the same time.

2 TASK DESCRIPTION

We consider the problem to discover predefined anomalies from thousands of free-text reports as a supervised learning problem where the algorithm classifies every free-text report as belonging to one or more of known anomaly categories. The assumption that a single report may fit several categories (predefined anomalies) brings our task to be so called multi-label text categorization. It may be performed independently for each category by means of One-Versus-Rest approach and after this one can select for report under investigation the most appropriate category or categories ("multi-class & one label" classification).

Classical Text Categorization algorithm includes the following steps:

Step 1: Preprocessing. The first task in this step is to represent text and to select features. The vector space model is used for the representation of the text documents. Each document may be represented as a vector of words. The entries in the vector are simple binary feature values, just because a word either occurs or does not occur in a current document, or the word occurrence frequency in a document. To reduce number of features (i.e. to control the vocabulary size) several approaches may be used, for example Stemming and Lemmatization. Stemming is a well known technique of the word reduction when common suffix and prefix are stripped from the original word form. Lemmatization is a process by which words are reduced to their canonical form (e.g., verbs – to their infinitive). Additional approach is the "Exclusion List". Exclusion list is a list that may include non-significant words such as "and",

'be", "about", etc. Removing these words may drastically reduce the system vocabulary size and therefore allows focusing only on important content words, thus improving the treatment efficiency. An additional approach may be used by eliminating words that appear in only certain number of documents. This number of documents (one, two, three, or more) depends on the specific implementation.

Step 2: Learning and Tuning of the Algorithm. This stage involves considering some limited amount of various text mining models and choosing the best one based on their predictive performance to produce stable results across documents, marked by the user. The goal of automatic text-categorization system is to assign not-marked documents to one or more of predefined categories on the basis of their textual content. Optimal categorization functions can be learned from labeled training examples (Training Set – get after real, human expertise, i.e. after expert marking a sub-set of documents). During text categorization, used by some Training Set, the following tasks should be solved:

- Optimal selection of weights of single kernels - it is search of some control parameters by means of Quadratic Programming Task solving;
- Optimal choice of meta-parameters ("meta-parameters choice" means choice of type and value of each of many parameters of kernels, penalty values) by means of cross-validation using.

Step 3: Deployment**.** This final stage involves using the developed algorithm (with selected and defined meta-parameters and kernels weights) for not-marked documents in order to generate their labels.

We consider data points, received after human expertise performing, of the form:

(**X**[1], **Y**[1]), (**X**[2], **Y**[2]) ,…, (**X**[n], **Y**[n]) where:

the **Y**[i] is a *k* dimensional vector (y[1, i],... , y[j, i],... ,y[k, i]), and y[j, i] either "1" or "−1" - this label denotes the category j to which the point **X**[i] belongs. Label "1" means, that document i belong for category j, label "-1" means, that document don't belong for category j;

each of **X**[i] is a *m* dimensional vector of the binary values [0 ; 1] or TF.IDF values (Joachims 1997). Index i = 1…n, where n is full amount of documents on Training Set, used for current text categorization. For using of binary coding the component q of m dimensional vector **X**[i] equals for 1, if q-th word from vocabulary is concluded on the document number i, otherwise this component equals for 0.

For coding of document according word frequency the component q of m-dimensional vector **X**[i] equals for TF.IDF of this q-th word from vocabulary in the document i. Index q = 1…m, where m is full amount of words on the current vocabulary for category j (j = 1…k).

Our method is based on SVM binary classification approach, i.e. for performing of the multi-label categorization we have really to solve binary categorization of type One-Versus-Rest k times. According to this we will consider below only single category and index j of the current category will be omitted.

For classification according to current category we view set {**X**[i]**,** y[i]**}** as training data, which denotes the correct classification which we would like the SVM to eventually distinguish, by means of the dividing hyperplane, which takes the form

$$y(\mathbf{X}) = \sum_{i=1}^{n} a[i]y[i]K(\mathbf{X}, \mathbf{X}[i]) + b,$$

where K(**X,** **X**[i]) is kernel function and b is bias.

The training is really followed by a Quadratic Programming Task solving: to find values a[1],…,a[n] to minimize

$$\sum_{i=1}^{n}\sum_{p=1}^{n} a[i]a[p]y[i]y[p]K(\mathbf{X}[i], \mathbf{X}[p]) - 2\sum_{i=1}^{n} a[i]$$

$$\text{s.t. } 0 \le a[i] \le C[i], \quad \sum_{i=1}^{n} a[i]y[i] = 0.$$

Kernel parameters (type, degree of polynomial, delta for Radial Basis Function - RBF, etc.) and penalty parameters C[i] are meta-parameters; they are defined by means of tuning performing (cross-validation using) for current category. Usually C[i] are same for all points i = 1..n. We have to use different values due to the following reason - training set for multi-label and multi-class text categorization tasks is highly imbalanced. For example, for some category it may consist on 20000 documents, marked as "negative" and only 200 documents, marked as "positive". According this, penalty parameters C[i] may get following values:

- C_{pos}, if current report **X**[i] belongs to the positive-marked current category;
- C_{neg}, if current report **X**[i] belongs to the negative-marked current category.

Kernel functions may be for example as following:

- Linear : $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}*\mathbf{x}')$
- Polynomial :

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}' + 1)^d$$

- RBF (Radial Basis Function):

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

For each non-marked document **X** it is calculated its value $y(\mathbf{X}) = \sum_{i=1}^{n} a[i]y[i]K(\mathbf{X}, \mathbf{X}[i]) + b$.

If $y(\mathbf{X}) \ge 0$, the non-marked document **X** is recognized as "Positive" for current category, otherwise as "Negative".

3 ADVANCED APPROACH

Classical tuning (meta-parameters optimization) is performed by means of maximization of some integrated criterion. For balanced data sets one can use the criterion of
Accuracy = (TP + TN)/(TP + TN + FP + FN) and it is clear, that for Break-Even Point, where Recall = Precision, the Accuracy ≈ Recall ≈ Precision.

For imbalanced data sets the widely used criteria is F-measure, which is defined as the harmonic mean between Recall and Precision:

F = 2/(1/Recall + 1/Precision). Usage of the F-measure to compare classifiers assumes that Precision and Recall are equally important for the application. If one criterion is more important than the other, then one should use the p-weighted harmonic mean: $F_p = (1 + p)(1/Recall + p/Precision)$, where p describes how much the Recall is more important than the Precision.

Also the wide used criterion is Break-Even Point.

Fig. 1 illustrates typical graph "Recall Versus Precision" for imbalanced data sets. From this graph one can see that it is impossible to support simultaneously high values both for Recall and Precision.

To support required high values for both Recall and Precision, following additional meta-parameters are introduced:

- G_{low} – Low boundary for separating function (i.e. for $y(\mathbf{X})$);
- G_{high} – High boundary for separating function.

Proposed mixed, partially automated text categorization algorithm is performed as following:

If $y(\mathbf{X}) \geq G_{high}$, the non-marked (new) document **X** is recognized as "current category" and expert should not verify this solution;

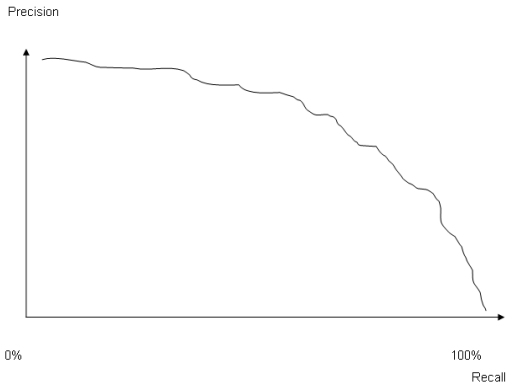

Figure 1. Typical graph "Recall Versus Precision"

If $y(\mathbf{X}) \leq G_{low}$, the non-marked document **X** isn't recognized as "current category" and expert should not verify this solution;

If $G_{low} < y(\mathbf{X}) < G_{high}$, the expert should manually verify this document for current category.

The following procedure is proposed:

1. Customer selects required values of Recall (REC_{req}) and Precision ($PREC_{req}$) per current category – e.g., 0.9 for Recall and 0.95 for Precision.

2. By means of cross-validation on the Training Set the modified tuning is performed (the meta-parameters are selected) s.t. Recall ≥ REC_{req} according for the separating hyperplane $y(\mathbf{X}) = G_{low}$ and Precision ≥ $PREC_{req}$ according for the separating hyperplane $y(\mathbf{X}) = G_{high}$. Due to supported high value of Recall, according for the separating hyperplane $y(\mathbf{X}) = G_{low}$ the obtained value of Precision may be too low, e.g. 0.2…0.3 and even less.

3. Automatic Text Categorization is performed for new documents on the Test Data Set.

4. Manual (Human) Expertise is performed for non-recognized part of documents, i.e. for **X** with $G_{low} < y(\mathbf{X}) < G_{high}$. It should not increase Recall value of current category recognition, obtained after Automatic Text Categorization, but should essentially increase Precision value for documents **X** with $y(\mathbf{X}) \geq G_{low}$.

Modified tuning procedure, based on cross-validation, is proposed to select optimal values of standard control meta-parameters (*Kernel type*: linear, polynomial, RBF; *Kernel parameters*: delta, degree; penalty parameters) and optimal values of proposed meta-parameters (G_{low} and G_{high}). The purpose of this tuning is:

- To support required Recall and Precision levels;
- To minimize amount of reports, which have to be verified by expert manually after automatic report categorization.

Proposed tuning procedure is two-staged:

1. Standard meta-parameters are defined as usually in the SVM method: some values are fixed, SVM Quadratic Programming Task is solved for current fold of Training Set, obtained values of a[1],…, a[n] are used for Validation Set classification, output criterion is calculated (mean value for all folds), meta-parameters' values are changed, etc.

2. Proposed meta-parameters G_{low} and G_{high} should be selected as fixed values of the standard meta-parameters just after SVM Quadratic Programming Task solving. From strict theoretical point of view this is not the most accurate solution, because bias b should be calculated from obtained values of parameters a[1],…,a[n], but practically - from many numerical experiments - we could observe that obtained approximate solution is near optimal. The said selected values of the G_{low} and G_{high} allow to perform tuning without additional solving the large-scale SVM Quadratic Programming Task.

To define proposed meta-parameter G_{low}, we take into account that both: "Amount of reports which have to be verified by expert manually" and "Recall" are monotonous non-increase functions of the variable G_{low}. Thus to support Recall ≥ REC_{req} and simultaneously to minimize "Amount of reports which have to be verified by expert manually", it is possible to use very fast method of secants.

To define proposed meta-parameter G_{high} we can use (as definition of standard meta-parameter values) a partial enumeration with some discrete steps.

It is not necessary to use exactly two proposed meta-parameters G_{low} and G_{high}. One can use only one meta-parameter G_{low} to support required Recall level. In this case G_{high} = Infinite and an expert manually checks all reports, recognized by automatic procedure as "pseudo-positive" (s.t. $y(\mathbf{X}) > G_{low}$). This case is very interesting: it supports zero value of FP after expert verifying, i.e. Precision = 1. Drawback of this approach (in comparison with "two additional meta-parameters" approach) is some increasing of the report amount to be verified by expert manually after automatic data categorization. Advantages of the "one additional meta-parameter" approach are as following:

- Faster tuning, because we should not select optimal value G_{high}
- It supports more accurate solutions for non i.d.d. (identically and independently distributed) situations (see chapter 5)

It is necessary to note, that for high-imbalanced data sets (with less than 3 % of positive samples) usually it is enough to use only "one additional meta-parameter" approach.

The reason is as following: due to small amount of the positive reports the amount of reports **X** with $y(\mathbf{X}) \geq G_{high}$ (the non-marked documents **X,** which are recognized as positive and should not be verified by expert) is negligible - see example in the next chapter. But for low-imbalanced data sets (i.e. amount of positive samples is more than 3 %) the amount of reports **X** with $y(\mathbf{X}) \geq G_{high}$ may be large and we should not ignore this amount – for such anomalies the approach with use of "two additional meta-parameters" is more preferable.

4 NUMERICAL EXAMPLE

The ASRS (http://asrs.arc.nasa.gov/index.html) On-Line Data Base was used in order to evaluate the proposed method empirically. This Data Base collects reports of the USA flights. There are similar data bases for flight reports collection in UK (CHIRP), France (REC), Japan (ASI-NET), Korea (KAIRS), etc.

The ASRS (Aviation Safety Reporting System) is a well-known textual data set for aviation safety. This data set is a collection of ~ 300,000 reports categorized into 58 different anomalies (categories). Examples of anomalies, extracted from ASRS data base, are "805: Spatial Deviation – Track or Heading Deviation" (occurs in 10% of the reports), "809: Altitude Deviation – Overshoot" (occurs in 7% of the reports), "817: Ground Incursion – Landing without Clearance" (occurs in 2% of the reports), "856: In-flight Encounter – Turbulence" (occurs in 3% of the reports), "859: In-flight Encounter – Weather" (occurs in 6% of the reports), "860: In-flight Encounter - VFR in IMC" (occurs in 1% of the reports), "896: Other Anomaly - Loss of Aircraft Control" (occurs in 4% of the reports), etc.

Each single report may be assigned to "no one" (zero) up to 10 such categories. We have extracted 10,000 reports as training data (Training Set) and next 10,000 reports as test data set. We removed vocabulary words included either in the stop list or in only one report. After this we have performed vocabulary reduction independently for each category up to 500 vocabulary words. Selection of the optimal values of meta-parameters (both standard and proposed G_{low} and G_{high}) was based on 2-fold cross-validation on 10,000 reports of the Training Set. Results for category "860: In-flight Encounter – VFR in IMC " are summarized on the following table ($REC_{req} = 0.9$, $PREC_{req} = 0.95$).

Just for a comparison let's consider results, obtained by standard SVM approach:

- Precision = Recall = 0. 47 - for tuning based on "Break-Even Point" criterion maximization;
- Precision = 0.26, Recall = 0.83, $F_5 = 0.61$ - for tuning based on "F_5" criterion maximization.

It is also remarkable to compare above results with results of usage of "one additional meta-parameter" approach (G_{high} = Infinitive):

- "Report amount which should be checked by expert" = 930,
- "Positive report amount from all checked reports" = 90,
- Recall = 0.9, Precision = 1.0,
- "Acceleration of Expert Work" = 10.8 times.

It can be shown that for comparison with "two additional meta-parameters" approach the reduction of acceleration is negligible.

For other anomalies, with low-imbalanced data sets, difference may be more essential. For example, for anomaly "801: Aircraft Equipment Problem - Critical" the report amount, predictive as positive and those which should not be checked by expert, will be large (= 860 reports) and so using the approach of "one additional meta-parameter" instead of "two additional meta-parameters" approach, we will get the essential reduction of the acceleration – from 7 times to 5 times.

Table 1 Results of report categorization

Optimal meta-parameter values, after Cross-Validation and Tuning on Training Set	
Kernel Type	RBF
Delta	0.01
C_{neg}	3
C_{pos}	60
G_{low}	-1.0
G_{high}	1.9
Results after Automatic Categorization on Test Set	
Report amount should be checked by expert	910

Report amount predictive as positive and should not be checked by expert	20
Amount of really positive of them (part of TP)	15
Amount of negative reports of them (FP)	5
Report amount predictive as negative and should not be checked by expert	9070
Amount of positive report of them (FN)	10
Final Results, after Human Expertise on 910 documents of Test Set	
Positive report amount from all checked reports	75
Amount of really positive reports (TP)	90 (= 75+15)
Full report amount predictive as positive (TP+FP)	95
Recall	0.9
Precision	0.95
Acceleration of Expert Work (reduction of report amount, verified by expert)	11.0 times(= 10000/910)

5 SOLUTIONS FOR NON-STABILITY OF THE INPUT STATISTICS

The above proposed approach allows us to take into account the possible non-stability of the input statistics. The majority of the machine learning algorithms assume that the data are identically and independently distributed (i.i.d.), but this may not be the actual situation. In real life often there is a lack of stability of the report statistical parameters, i.e. the frequency of words on the report is changed significantly – due to new word appearance, new report writers appearance, etc. So if we use same training set for all possible test sets in the future, we'll use it with a test set drawn from a different distribution to the training set.

For some tasks it is easy and cost-effectively to get new "marked" training set for current test set, but in some cases it requires a lot of resources. Examples of first type of tasks are weather prediction, inflation prediction, etc., where after some event appearance the exact category of data is assigned without bringing too much human expertise. Examples of second type of tasks are Image Recognition, Handwritten Documents Categorization, Aviation Safety Reports Categorization, etc., for which the exact category of data may be get only manually, using human experts for reading the training set documents and classifying ("marking") them. In this case the problem is firstly that this expertise is expensive or difficult to obtain, and secondly that such a human labeling of data can cost both time and money.

As a result, for non-i.d.d. data sets, we should perform manual marking of the full current training set, i.e. to carry out human expertise on very large amount of reports. We propose another approach, based on the algorithm described in Chapter 3 – without additional human expertise - using directly as labeled data only the reports, marked by experts during classification of the non-recognized part of the documents (after the Automatic Text Categorization). Accordingly the above proposed methodology is a typical SSL (semi-supervised learning).

The SSL methods are well known when the training set consists of significant quantity of un-marked reports and a small portion of marked documents (Chapelle et al. 2006, Nigam et al. 2000, Wang et al. 2009). These methods are generic, in some sense they combine clustering (un-supervised) and classification (supervised) methods.

We developed a much simpler algorithm essentially using specific features of our task: imbalanced type of data sets and high requirements for Recall and Precision. To increase accuracy, for non-i.d.d. data sets we will use only "one additional meta-parameter" approach (i.e. G_{high} = Infinite).

Consider example from chapter 4 and will use Test_Set = Reports [10001: 20000] as Training Set for next Test_Set = Reports [20001: 30000].

Based on "one additional meta-parameter" approach we've got following estimations obtained after first automatic and then partial categorization by experts:

- 9070 reports were automatically recognized as negative, 10 of them really were positive;
- 90 reports were true recognized by expert as positive, 840 reports were true recognized by expert as negative.

Taking in account 9910 negative reports (9910 = 9070 + 840) the influence of wrong classification of 10 positive reports is negligible – less than 0.1%. Nevertheless for the 90 positive reports the influence of real loss of the additional 10 positive reports is essential.

To compensate this loss, it is necessary to increase amount of positive report for 10 reports and to reduce amount of negative reports for 10 reports. Certainly, we don't know this amount exactly, but it is $\approx 10000*FN_{valid}/AM_{valid}$,

where FN_{valid} is an average FN value, obtained for validation sets during cross-validation on initial Training_Set = Reports [1:10000],

AM_{valid} – amount of the reports on the single Validation Set (in our case, for 2-fold cross-validation, $AM_{valid} = 5000$).

Note, that the initial Training Data Set is fully labeled by means of human expertise, so values of FN_{valid} and AM_{valid} are measured directly. In next iteration the Training Data Set will be Reports [20001:30000], it is also fully labeled. We can calculate how many positive reports to increase and how many negative reports to decrease from this Data Set according to expression $10000*FN_{valid}/AM_{valid}$,

where FN_{valid} is average FN value, obtained for validation sets during cross-validation on Training_Set = Reports[10001:20000], etc.

We can randomly select reports of calculated amount from 9070 negative reports (for pruning) and randomly select reports of calculated amount from 90 positive reports (for duplication). Moreover, we

can perform choice of these reports more correctly - to prune the "most problematic for recognition" 10 negative and to duplicate 10 positive reports as following:

- reports, placed nearest to separation line, i.e. which have values of goal classification function y(**X**) with minimum difference from value G_{low} = -1;
- reports, which have minimum probabilities to be negative and positive (about probabilities calculating see : Niculescu-Mizil & Caruana 2005, Platt 1999).

From this consideration it is possible to use for current Test Set categorization previous reports as Training Set with labels, obtained by previously automatic categorization with small amount of human expertise, and some modifications according to the above proposed rules.

To increase accuracy of the Test Set report categorization, it is recommended to reduce Test Set size. This solution will allow us to take into account the changing of report word frequencies more correctly. Size of 10000 reports for Test Set is too large, it corresponds to ~ 1 year statistics. For non-i.d.d. report statistics the appropriate solution may be Test Set = 1000 reports. In this case we don't have to calculate how many positive reports to increase and how many negative reports to decrease for the full Training Data Set - it is only necessary to take new Training data set of approximately 1000 reports.

6 CONLUSIONS

This paper presents a novel supervised learning algorithm making possible an efficient study of safety and reliability problems reported from the field as a free text by pilots, operators, inspectors etc. The presented semi - automated approach makes feasible to find most of field anomalies automatically, by text categorization algorithm mixed with reasonable and cost-effective amount of human expertise. It focuses on selecting best possible and most informative examples for manual labeling. Proposed approach also allows to take into account non-stability of the report statistics making able for safety professional to get much better results than using the traditional algorithms - providing high values of output criteria (e.g., both Recall and Precision have to be simultaneously more than 90…95 %).

The effectiveness of the presented methodology was successfully demonstrated by extensive large-scale categorization work performed the aerospace safety ASRS data base.

REFERENCES

Akbani, R., Kwek, S. & Japkowicz, N. 2004. Applying support vector machines to imbalanced datasets. In: *Machine Learning, ECML 2004*: 39-50.

Chapelle, O., Zien, A., & Scholkopf, B. (Eds.). 2006: *Semi-supervised learning*. MIT Press. Cambridge, MA.

Chawla, N.V., Bowyer, K.W., Hall, L.O., & Kegelmeyer, W.P. 2002. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.*16 : 321–357.

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

Joachims, T. 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceeings of ICML- 97, 14th International Conference on Machine Learning. Nashville, 1997*: 143–151.

Imam T. , Kai Ming T. & Kamruzzaman J., 2006. Z-SVM: An SVM for improved classification of imbalanced data. *Lecture Notes in Computer Science* 43: 264-273.

Lewis, D. D. 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. In *Proceedings of ECML-98, 10th European Conference on Machine Learning. Chemnitz, Germany*: 4–15.

Niculescu-Mizil A. & Caruana R. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22-th International Conference on Machine Learning, Bonn, Germany.*

Nigam, K., Mccallum, A. K., Thrun S. & Mitchell, T. M. 2000. Text classification from labeled and unlabeled documents using EM. *Mach.Learn. 39*, 2(3): 103–134.

Platt, J. 1999. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers:* 61–74.

Rocchio, J. 1971. Relevance feedback in information retrieval. In Salton (ed.), *The SMART Retrieval System: Experiments In Automatic Document Processing, Chapter 14*: 313-323, Prentice-Hall.

Scholkopf, B. & Smola, A.J. 2001. Learning with kernels: support vector machines, regularization, optimization, and beyond. The MIT Press.

Srivastava, A.N., Akella, R., Diev, V., Kumaresan, S. P., McIntosh, D. M., Pontikakis, E. D., Xu, Z. & Zhang, Y. 2006. Enabling the discovery of recurring anomalies in aerospace problem reports using high-dimensional clustering techniques. In *Proc. of IEEE Aerospace Conference, 24 Jul 2006.*

Wang J., Shen X. & Pan W. 2009. On Efficient Large Margin Semi-supervised Learning: Method and Theory. *Journal of Machine Learning Research* 10: 719-742.

Wiener, E. D., Pedersen , J. O. & Weigend, A. S. 1995. A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval. Las Vegas, NV, 1995*: 317–332.

Wu, G. & Chang, E.Y. 2005. KBA: Kernel boundary alignment considering imbalanced data distribution. *IEEE Trans. Knowl. Data Eng.* 17(6):786–795