Maximizing classifier yield for a given accuracy

Wessel Kraaij [a] Stephan Raaijmakers [a] Paul Elzinga [b]

[a] *TNO Information and Communication Technology*
Delft, The Netherlands
[b] *Regional Policeforce Amsterdam Amstelland*
Amsterdam, The Netherlands

Abstract

We propose a novel and intuitive way to quantify the utility of a classifier in cases where automatic classification is deployed as partial replacement of human effort, but accuracy requirements exceed the capabilities of the classifier at hand. In our approach, a binary classifier is combined with a meta-classifier mapping all decisions of the first classifier that do not meet a pre-specified confidence level to a third category: *for manual inspection*. This ternary classifier can now be evaluated in terms of its *yield*, where yield is defined as the proportion of observations that can be classified automatically with a pre-specified minimum accuracy.

1 Introduction

The evaluation practice of information processing tasks such as classification, detection and ranking is a non-trivial issue, where no ideal recipe exists. Evaluation is either tailored toward component benchmarking or can be focused on end-to-end user experience. The component evaluations have their roots in the Cranfield Information Retrieval experiments that were a model for the successful TREC evaluations [10]. These batch style experiments have for a long time focused on automatic only experiments, where human involvement is separated as much as possible from the actual experiments in order to avoid inter user variability and completely focus on the actual system component under scrutiny. Such batch style experiments have been attractive for IR researchers and even inspired evaluations in other communities such as natural language processing, since experiments were easy to conduct, and also very economic because humans were excluded from the loop (except for creating the ground truth). Still many researchers felt that these studies were limited, since they failed to model a real search process.

The component based evaluation which is the model for TREC is sometimes referred to as intrinsic evaluation in contrast to an evaluation where the component's performance is measured in the user context (extrinsic). When evaluating a complete system, intrinsic evaluation approximates *performance* evaluation and extrinsic evaluation is related to *adequacy* measurement [4]. In such a task based evaluation, factors such as usability play a crucial role. Performance measurements are usually aimed at comparing systems, whereas adequacy measurements focus more on the usability and practical use for an end user.

In many scenarios, the classification accuracy of a machine learning based classification system is not sufficiently high, since the tasks at hand are difficult. We propose that for these scenarios, systems can still successfully be deployed if only the "easy cases" are classified automatically. In such a deployment scenario, quality standards can still be met, whilst reducing (and not completely replacing) the manual workload.

The objectives of this paper are two-fold:

1. Introduce a novel ensemble of classifier evaluation measures which can evaluate the deployment of a classifier which only partially replaces human labeling.

2. Develop a ternary classifier that can operate at a pre-specified accuracy by forwarding "difficult" items for manual processing.

	assigned class: ”+”	assigned class: ”–”
ground truth: ”+”	TP	FN
ground truth: ”–”	FP	TN

Table 1: Classification contingency table. Precision is defined as $TP/(TP + FP)$ and recall is defined as $TP/(TP + FN)$.

In this paper we propose a novel ensemble of evaluation measures for classification tasks that can be used for component evaluations. The distinguishing characteristic of this new ensemble is the fact that both measures (accuracy and yield) are motivated from the task viewpoint and directly relate to potential cost savings in terms of reduced manpower. The structure of this paper is as follows: in section 2 we give a formal definition of the new ensemble of evaluation measures and discuss the relationship of these measures with operational characteristics of an abstracted workflow (an office where analysts manually label documents). Section 3 illustrates the ensemble of measures by reporting experiments concerning automatic detection of domestic violence cases in police files and a spam detection task. Section 4 describes the ternary classifier architecture. Section 5 presents two experiments that illustrate the value of the evaluation method and the ternary classifier. The paper concludes with a discussion section.

2 Classifier accuracy and classifier yield

Several evaluation measures dominate the field of component based evaluation for classification and ranking tasks. The field of information retrieval evaluation popularized the precision and recall measures. These are set based measures which can best be visualized by looking at a contingency table (Table 2). Whereas the original precision and recall measures are hardly used anymore in IR (instead mean average uninterpolated precision is the norm for ranking tasks), they are regularly reported for classification experiments. Precision and recall have the desirable property that they relate well to intuitive characteristics of quality. Better systems have higher precision and or recall values. A disadvantage of precision and recall is that the test set must be a representative sample of the real class population. An opposite approach is to quantify the error rates of a classifier, where a better system has smaller error rates. For a binary classifier scenario both type I and type II error rates (false alarms and misses) can be measured independently from the actual class distribution in the test set.

Precision is a measure of fidelity and is inversely related to type I errors (false positives). Recall can be seen as a measure of completeness, being inversely related to type II errors (false negatives). An important nuance to make here is that fidelity and completeness are defined with respect to the positive class label, i.e. the task modeled is correctly identifying items with a positive class label. Precision and recall can be combined into a single measure F_β [9], which helps to compare systems at a certain operating point (usually precision and recall are considered equally important). Note that precision and recall are defined from the perspective of the positive class.

Another measure that is often reported for classifier evaluation experiments is classifier accuracy. This is an intuitive measure for classification quality provided the class prior probabilities do not differ too much. The accuracy quantifies the quality of both positive and negative decisions made by the (binary) classifier. This averaging behaviour makes accuracy highly sensitive to a skewed distribution of class priors (imbalanced natural class distribution). This means that it is difficult to interpret accuracy results unless the class distribution of the test set is known. A simple majority classifier can have a very high accuracy for skewed distributions.

A subclass of typical real-life classification problems are detection tasks. These can be characterized as binary classification tasks with a skewed natural class distribution i.e. the negative cases are much more common than the positive cases. We are aware of the problems that these kinds of tasks pose for training classifiers and for designing benchmark data sets (some of these issues were briefly introduced above). A training data set needs to contain sufficient positive examples of a relatively rare phenomenon. The test data set however should contain enough negative examples in order to have a proper estimate of false positives. These are all important issues for the design of evaluations, but they are not the focus of this paper. Our claim is that just stating that a classifier has a certain F_1 value or accuracy cannot be translated in terms of its potential for operational deployment. Also, in some scenarios the problem is so difficult that state of

	assigned class: "+"	assigned class: "–"	assigned class: "?"
ground truth: "+"	TP	FN	M
ground truth: "–"	FP	TN	

Table 2: Classification contingency table for the ternary classifier

the art classifiers do not meet the minimum quality requirements that have been defined for this task. Still, if we could modify the workflow of human analysts and the classifier architecture in such a way that part of their work could be automated, while meeting the minimum quality requirements, it is easy to define a business case. We therefore propose a novel and intuitive way to quantify the utility of a classifier in cases where classification is applied in order to partially replace human labour, but accuracy requirements exceed the capabilities of the classifier at hand. Typical application scenarios are binary detectors. In our approach, a binary classifier is combined with a meta-classifier mapping all decisions of the first classifier that do not meet a pre-specified confidence value to a third category: *for manual inspection*. The classifier combination can be seen as a ternary classifier, which can now be evaluated in terms of its *yield* at a pre-specified confidence level, where yield is defined as the proportion of observations that can be classified automatically with a minimum pre-specified accuracy. In a way, accuracy and yield model the same intuitive aspects that underly precision and recall, classifier accuracy is a way to measure the fidelity of the classification task and classifier yield can be viewed as a measure for classifier completeness at the task level. The intended use of the ensemble $\{$*accuracy,yield*$\}$ is to measure the classifier yield at a fixed (minimum) level of accuracy. As an example, we could be interested in the yield of a biometric detector at an accuracy level of 99%.

Table 2 shows a modified contingency table where the classifier can assign one additional label: "?" (queue for manual inspection). Now accuracy can be defined as usual:

$$\textit{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

and yield can be defined as:

$$\textit{yield} = \frac{TP + TN + FP + FN}{TP + TN + FP + FN + M} \tag{2}$$

It is easy to see that the classifier yield is just the proportion of observations that is not labeled as M.

3 Related work

As far as we know, the proposed ensemble of measures (yield at minimum accuracy) is a novel way of measuring the quality of a classifier. There are several established evaluation traditions that have some elements in common. The TREC filtering task used a linear utility function for the adaptive filtering task, which is a rather complex classification task where a system can use feedback in order to set its optimal operating point (decision threshold) in a dynamic fashion. The linear utility is defined as [5]:

$$\textit{linear utility} = \alpha \times TP + \beta \times FP + \gamma \times FN + \delta \times TN \tag{3}$$

This is essentially a cost function, where parameters must be chosen to model a particular user scenario. Choosing four parameters (which can be negative) is non-trivial, and therefore in our view not so intuitive. Linear utility could be extended to handle the five-cell contingency table corresponding to our ternary classifier, but that would mean five parameters to choose. A more elegant way to model the *cost* of running a certain classifier on a dataset is the family of cost functions that were developed in the Topic Detection and Tracking (TDT) framework [3]. The basic cost function is defined as follows:

$$\textit{detection cost} = C_{Miss} \times P_{Miss} \times P_T + C_{FA} P_{NT} P_{FA} \tag{4}$$

where C_{Miss} and C_{FA} are fixed cost parameters that tax type II and type I errors respectively, P_{Miss} and P_{FA} are the probabilities (normalized counts) of type II and type I errors (false alarms), and $P_T = 1 - P_{NT}$ is the prior probability of a positive class label (*T=target*). Usually, the detection cost is measured at different levels of Miss/False Alarm trade-off by threshold sweeping, thus generating a detection cost curve. The detection cost function is motivated by the desire to quantify different types of error and sum the complete cost of a detection task for a certain data collection (taking into account the relative proportion of the class

population sizes). However, the detection cost is based on a fully automatic scenario. Incorporating the cost of manually assessing observations would make the detection cost function less intuitive.

Another common aggregate statistic for measuring classification is the AUC (area under (ROC) curve) [2]. AUC is the ROC (receiver operating curve) equivalent of mean average uninterpolated precision. ROC is based on a plot of the true positive rate (recall) versus the false positive rate. ROC curves are less optimal for unbalanced classes, since the interesting part of the curve needs zooming [6]. In principle it should be possible to use our ternary classifier architecture for a yield fixed AUC evaluation scenario, although AUC is not a very intuitive quality measure for non-experts.

Finally, a common evaluation procedure for biometric detectors is to measure the false alarm rate (FAR) at a fixed maximum false reject (miss) rate (FRR) or vice versa [1]. Our proposed procedure is similar in the respect that a certain operating point is pre-defined in order to compare systems. The pre-defined operating point provides an "anchor" in the recall-precision trade-off and simplifies evaluation to a single measure just like F_β defines a certain operating point in the precision recall space.

4 An example ternary classifier

The experiments that were carried out to illustrate the evaluation procedure were based on a two-level classifier architecture. The first level classifier was implemented by an information diffusion kernel machine. This kernel machine presupposes L1-normalized data (relative frequencies) and estimates similarity between documents using a geodesic distance measure applied to the Riemannian manifold that represents this data [11]. The (parameter free) diffusion kernel machine was modified to provide a posterior probability as output in addition to the predicted class [7]. The mapping function was trained on a separate development data set. The posterior probability (Platt score) was subsequently used as an input score σ for a meta-classifier that was implemented by a decision rule based on two thresholds θ_l and θ_u. The decision rule was defined as follows:

$$\textit{prediction}(\sigma) = \begin{cases} + & \textit{if} \quad \sigma => \theta_u \\ M & \textit{if} \quad \theta_l < \sigma < \theta_u \\ - & \textit{if} \quad \sigma <= \theta_l \end{cases} \tag{5}$$

The thresholds maximizing the yield while satisfying the pre-specified minimum accuracy were computed through exhaustive search by a two dimensional parameter sweep (for both threshold parameters θ_u and θ_l) on a development set.

The development data set for parameter training should be chosen carefully since we assume that the class distribution is the same in the development set and the test set and that the Platt score distribution is more or less similar in the development and test set, for both classes.

5 Experiments

We will illustrate the use of the evaluation procedure by two experiments. The first experiment concerns the detection of domestic violence in police files. The second experiment is about spam detection

5.1 Detection of domestic violence

Taking adequate action in cased of domestic violence is one of the focal points of the regional police force Amsterdam-Amstelland (RPAA). Recognition of domestic violence as such in incident reports is not an easy task, since domestic violence has a complex legal definition where several conditions need to be checked. Domestic violence is not always marked as such in the reports by the registrating police officer, so it is desirable to recognize these cases post-hoc automatically. The current practice for filtering out domestic violence cases from the full database of incident reports is based on a rule based system. Rules are created and maintained manually. Unfortunately the current rule set creates a very high number of false positives, which means that all filtered cases currently are subjected to a manual check. In order to minimize the number of manual checks, two classifiers were compared on site. A baseline rule based classifier[1] using hand-crafted thesauri (more elaborate and refined than the incident-report filtering system) and the ternary

[1]This classifier is actually a ranking system, where a decision threshold was chosen manually.

	accuracy	yield
baseline classifier	0.73	1
diffusion kernel machine	0.84	1

Table 3: Results for the detection of domestic violence on the full test set using a single classifier

	accuracy	yield
development set	0.90	0.70
full test set	0.92	0.86
test set sample A	0.93	0.86
test set sample B	0.92	0.89
test set sample C	0.93	0.86

Table 4: Results for the detection of domestic violence experiment using the ternary classifier

classifier discussed in Section 4. The ternary classifier architecture used the same feature set as the baseline classifier. Example features are *my father beats* and *my uncle abducts*, where verb forms were normalized.

The evaluation procedure based on accuracy and yield was applied in order to provide simple intuitive statistics that would enable a transparent interpretation of what a deployment of an automatic classifier would mean in terms of reduction of processing time, whilst maintaining the required quality level.

The following datasets were used:

training set A collection of 1736 reports, manually re-checked. 1101 positive cases. A random sample of 200 case files was used for development, the rest (1536) for training.

test set A held out collection of 2291 reports, labeled by registrating officer. 541 positive cases

As a first step the diffusion kernel and Platt function were trained on the development set. In a second step, optimal upper and lower decision score threshold were computed using the development data with a pre-specified accuracy > 0.90.

Table 3 lists the evaluation results (measured in terms of accuracy) for the baseline rule based ranking classifier and the diffusion kernel machine. The more advanced classifier architecture has a superior performance thanks to its generalizing capabilities. Still the accuracy of the diffusion kernel machine is too low for deployment at RPAA. In a second step, score thresholds are learned on a development set[2] to isolate those reports where the classifier decision is based on a low confidence score. These reports can then be forwarded for manual inspection. As an illustration, Figure 1 shows the probability that the classifier is correct as a function of its score.

The important question is whether decision thresholds can be learned and whether they are robust. Table 4 lists the accuracy and yield of the ternary classifier for development and test sets. As an additional diagnostic, three random samples of the test set (sample size = 1000) were evaluated. The obtained accuracy and yield on the test set are both higher than on the development. This could be explained by the fact that the test set was obtained from cases from a different year, where annotation standards might have changed. Still, results of the classifier on development and test set show the potential of the proposed approach, which seeks to minimize the amount of human labeling while meeting pre-specified quality standards. The results at various subsamples demonstrate the robustness of the parameter settings. Related work on the same dataset explores the possibility of involving a human expert for an interactive selection and definition of complex features, based on formal concept analysis [8].

5.2 Spam detection

As a second experiment we chose a spam detection task, available from the ECML 2006 Discover Challenge `http://www.ecmlpkdd2006.org/challenge.html`. The challenge consists of two separate tasks: a task (A) with many user-specific training data addressing user-specificity of the found solution, and a task (B) with a limited amount of data per user, addressing generalization over users. In this work, we

[2]We did some preliminary experiments varying the size of the development set and a size of 100 was still sufficient.

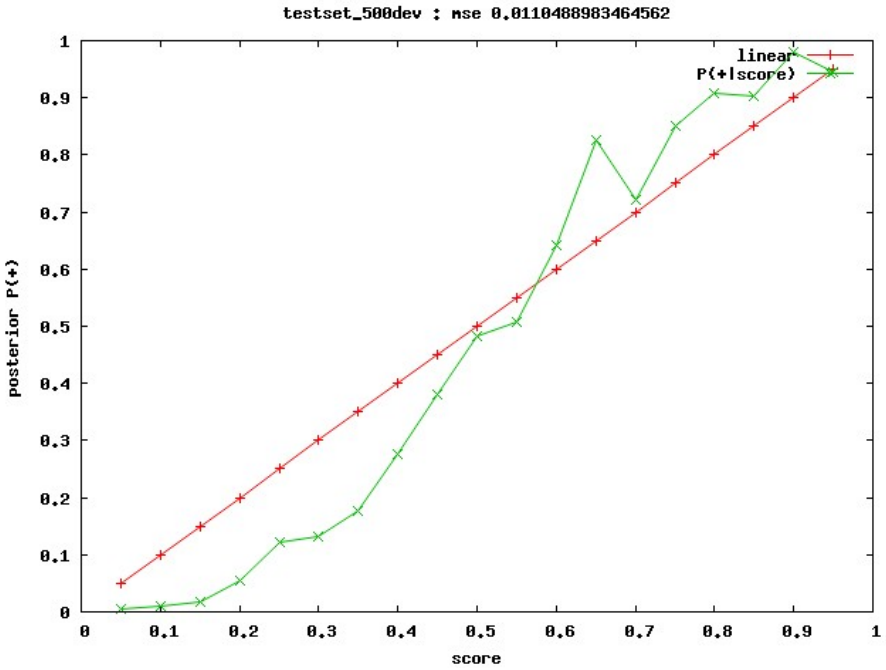

Figure 1: Posterior probability as a function of Platt score

	#pos dev	#pos test	binary classifier accuracy	ternary classifier accuracy	ternary classifier yield
user 0	248	1002	0.62	0.89	0.19
user 1	241	1009	0.65	0.90	0.39
user 2	268	982	0.78	0.91	0.69

Table 5: Results for the detection of spam emails using a binary and ternary classifier

limit ourselves to task A. All data sets consist of word/frequency pairs, which can be easily normalized to L1.

Task A models three users. For each user there are 4000 labeled training email messages and 2500 for evaluation. We divided the evaluation sets in a development set of 500 emails and the remaining 2000 for evaluation.

Table 5 lists the results of the spam detection experiment. The first two columns give the number of spam messages in the development and test set respectively. The third column gives the accuracy of the standard binary classifier (diffusion kernel machine). The fourth and fifth column give results on accuracy and yield when the ternary classifier's thresholds have been set for a minimum accuracy level of 0.90 using the development subsets. The desired accuracy (0.9) can be achieved for about 20-70% of the email messages depending on the user, making it a much harder task than the domestic violence detection.

Figure 2 illustrates the optimal operation curves for each user mailbox in a so-called *yieldplot*, where the classifier yield is plotted as a function of the desired accuracy level.

6 Discussion and Conclusions

We have presented a new ensemble of evaluation measures for a setting where a classifier is used to partially replace human labelling effort. The measures accuracy and yield relate well to a more extrinsic view on evaluation, where the focus is on cost savings. Accuracy and yield can be seen as workflow oriented measures for 'fidelity' and 'completeness'. The simplicity of this approach does have some shortcomings. Indeed accuracy as an aggregated measure hides the different sources of classification quality. it is well known that accuracy is sensitive to class imbalance. An alternative ensemble based on false alarm rate, false reject rate

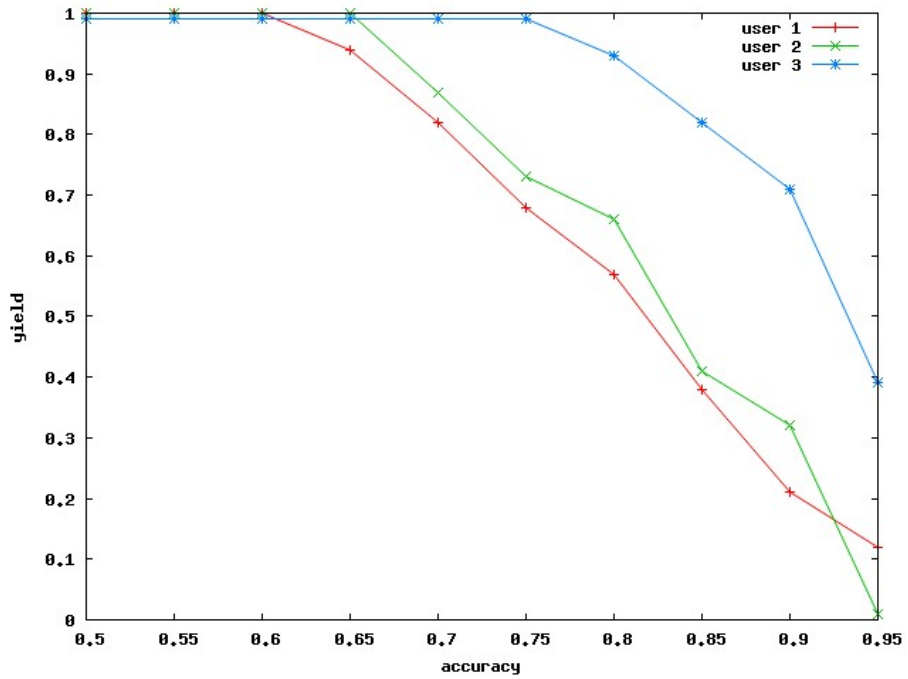

Figure 2: Yield as a function of (minimum) classifier accuracy, in the ternary classifier setting

and yield would solve this problem. However, this ensemble might be less intuitive for non-experts.

A second contribution of this paper is the concept of a ternary classifier, which forwards cases that cannot be classified with a pre-specified confidence to a human expert, thereby reducing the error rate of the classifier. Our method estimated two posterior probability threshold levels. The experiments show that the yield vs. accuracy plot makes it easy to use the ternary classifier in an operational workflow. Also, the ternary classifier can effectively forward difficult cases for human inspection.

In fact it is not essential that the classifier outputs true probabilities, it can be any monotonous increasing ranking function. As long as ranking values can be compared across collections, since the threshold values will always be optimized on a different data set than the test set.

There are several ways in which we plan to extend this research. We intend to look at the suitability of other (first level) classifier architectures, look at an ensemble of measures that makes a distinction between type I and type II error rates, and perform a more thorough analysis of the robustness of our parameter setting procedure.

References

[1] Ruud Bolle, Jonathan Connell, Sharanthchandra Pankanti, Nalini Ratha, and Andrew Senior. *Guide to Biometrics*. SpringerVerlag, 2003.

[2] Tom Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, 2006.

[3] Jonathan G. Fiscus and George R. Doddington. Topic detection and tracking evaluation overview. In *Topic detection and tracking: event-based information organization*, pages 17–31. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[4] L. Hirschman and H. S. Thompson. *Survey of the State of the Art in Human Language Technology*, chapter 13.1 Overview of Evaluation in Speech and Natural Language Processing. 1996.

[5] David A. Hull and Stephen E. Robertson. The TREC-8 filtering track final report. In *Proceedings of TREC-8*, 1999.

[6] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[7] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A.J. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans, editors, *Advances in Large Margin Classiers*, pages 61–74, 2000.

[8] Jonas Poelmans, Paul Elzinga, Stijn Viaene, and Guido Dedene. An exploration into the power of formal concept analysis for domestic violence analysis. In Petra Perner, editor, *ICDM*, volume 5077 of *Lecture Notes in Computer Science*, pages 404–416. Springer, 2008.

[9] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.

[10] Ellen M. Voorhees and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press, 2005.

[11] Dell Zhang, Xi Chen, and Wee Sun Lee. Text classification with kernels on the multinomial manifold. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 266–273, New York, NY, USA, 2005. ACM.