Information Filtering in Changing Domains

Carsten Lanquillon
DaimlerChrysler Research and Technology
D-89013 Ulm, Germany
Tel: +49 731 505 2809 Fax: +49 731 505 4210
carsten.lanquillon@daimlerchrysler.com

Abstract

The task of information filtering is to classify documents from a stream into either relevant or irrelevant according to a particular user interest with the objective to reduce information load. When using an information filter in an environment that is changing as time proceeds, methods for adapting the filter should be considered in order to retain the desired accuracy in classification. We favor a methodology that attempts to detect changes and adapts the information filter only if inevitable in order to minimize the amount of user feedback for providing new training data. Nevertheless, detecting changes may require costly user feedback as well. This paper explores a method for detecting changes without user feedback and briefly discusses strategies for adapting information filters. Empirical results with two simulated change scenarios based on real-world text data show that our adaptive information filters perform well in changing domains even without user feedback.

1 Introduction

The primary task of *information filtering* is to reduce a user's information load with respect to his or her interest. The filter is supposed to remove all irrelevant documents from an incoming stream, such that only relevant documents are presented to the user. Ranking these objects according to their degree of relevance is not required. Instead, all objects are classified immediately (on-line) as they arrive. Thus, information filtering can be described as a binary classification problem. In this paper, we only consider text documents and regard information filtering as a specific instance of text categorization.

Classification problems can be solved by applying supervised learning techniques which learn from a set of given examples and can then be used to determine the class of new, unseen observations. A comparison of some common learning algorithms for text categorization is given in [Dumais *et al.*, 1998]. The authors emphasize that modeling classifiers in static domains is generally sufficiently well controlled. The focus of this paper is to cope with dynamically changing domains.

The application of supervised learning algorithms for classification problems is based on the essential assumption that the distributions of the training data and the new data are somewhat similar. Even if this assumption may hold at first, it may become invalid in a long-term application. It should rather be expected that the content of the incoming texts changes as time proceeds. In this case, it is inevitable to adapt the applied information filter to the new situation in order to retain the desired accuracy in classification.

We favor a methodology that attempts to detect changes and adapts the information filter only if inevitable in order to minimize the amount of required user feedback. Alternatively, an information filter could be relearned in regular intervals no matter if changes really occur. This, however, requires the user to regularly provide new training texts and thus to read even many irrelevant texts. This is prohibitive since the task of information filtering is to reduce information load. Yet, we will use this approach as a benchmark.

The following section lists some related work and states the differences to this paper. Section 3 describes the types of changes we are looking at and in which way they effect classification performance. A method for detecting changes without any extra cost is explored in section 4, and section 5 briefly discusses two adaptation strategies. Some experimental results and a final discussion conclude this paper.

2 Related Work

Lewis did some seminal work on autonomous text classification system (see [Lewis, 1995]). His approach is designed for classifiers which respond to new documents with probabilities of class membership. He estimates the system's performance based on these probabilities and the classifier's decision about the class membership and thus does not require the true class labels to be specified. Lewis also suggests to monitor the performance in an ongoing fashion and adapt to changes if necessary. The method proposed in our paper uses classification confidences which are not assumed to be probabilities and monitors the fraction of decisions made with confidences below a given threshold. In statistical quality control terminology, this corresponds to observing the fraction of *non-conformities* with respect to classification confidence. Although no experimental results are given for dynamically changing text stream, extending the work of Lewis to other classification approaches seems very promising and should be pursued.

Concerning the detection of changes, the objective of this paper is similar to the task of *topic detection and tracking (TDT)* (for details, see [Allan *et al.*, 1998]). Yet, TDT is defined as an unsupervised learning problem. Hence, there are no relevance classes with respect to a particular user interest. Furthermore, an important issue in this paper is to minimize the amount of extra work to be done by the user. Therefore, the primary interest is not to detect the occurrence of changes as early as possible but to detect changes when they have a significant effect on the performance of the classifier.

Klinkenberg already did some interesting research on adaptive information filtering and essentially provided the basis for this paper (see [Klinkenberg and Renz, 1998]). However, while Klinkenberg focuses on changes in the user interest, we examine changes in the content of a text stream. The effect on the performance of an information filter is very similar, though. Klinkenberg also tries to cope with dynamic aspects by detecting changes first. He monitors indicators which are based on classification results and generally require the true class labels of the new texts in order to be evaluated. Although his adaptive approaches achieve promising results, in our setting they are intractable because providing true class labels for all new texts which have been kept from the user is prohibitive.

3 Dynamic Aspects

For the sake of simplicity, we assume that each text can be uniquely associated with exactly one topic. Further, each topic must be either relevant or irrelevant with respect to a particular user interest. Looking at a stream of incoming texts, we consider the following types of changes:

- new topics arise
- existing topics disappear
- existing topics change (i.e. the content).

A changing topic can be interpreted as the superposition of two similar topics, and one of these topics disappears while the other arises. Furthermore, a change due to a disappearing topic does not have any serious effect on the performance unless this topic is similar to an existing topic but belongs to the other relevance class or unless too many obsolete topics make modeling the information filter difficult. Therefore, in the following we will focus on changes due to new topics only.

Assuming forced recognition, a text of a new topic can either be classified correctly or be misclassified. If it is already classified correctly, the new topic is hard to notice at all. But with regard to classification performance it is not even necessary. An irrelevant text that is classified as relevant is rather easy to detect if we assume feedback for all texts *presented* to the user. However, it is difficult to realize that a relevant text is erroneously kept from the user. This problem is crucial to the application of information filters. How can be guaranteed that relevant texts are not permanently kept from the user?

Even if the quality may be assessable in static domains, its estimation may become obsolete in dynamically changing domains. Thus, it is inevitable to continuously observe the quality of the information filter. However, evaluating classification results in order to detect changes commonly requires knowledge of the true class labels. Providing these constitutes a lot of extra cost which conflicts with the objective of information filtering because texts that were kept from the user must then be read nonetheless.

4 Detecting Changes

This section explores a method for detecting changes without any extra cost. However, if a change is detected and it is advisable to adapt the classifier, extra work for providing new training examples may still be required.

For detecting changes, we assume that the text stream is divided into batches with respect to its chronological order. Even if information filtering is regarded as an on-line process where each text is classified as it arrives, constructing batches is quite natural since, for example, all texts arriving in the course of a day or a week can be grouped together. The value of any indicator that could be used to detect changes will then be calculated separately for each batch. Using batches offers the advantage to detect changes by observing the deviation from the indicator value of the current batch to the mean derived from indicator values of past batches while averaging and thus reducing stochastic perturbations and not paying to much attention to outliers. In the following we assume that a change has occurred if the current indicator value is above the mean indicator plus three times the standard deviation according to the *three-sigma control limit* of a *Shewhart control chart* (see [Lanquillon, 1998] for further details and [Montgomery, 1997] for an introduction to statistical quality control).

We consider indicators of the following two types:

- *Text properties*: The indicator characterizes a current subset of the text stream, e.g. class distribution or frequencies of words.
- *Classification properties*: The indicator is based on final or intermediate classification results, e.g. performance measures such as the error rate.

When calculating indicators derived from text properties based on new texts, there is a chance of detecting changes even before classifying these new texts. Thus a user can be warned that the results of the following classification may be uncertain. These indicators may prove useful for detecting changes, and we will explore them in future research. In the following, however, we focus on indicators which are derived from classification properties.

Common performance measures in information retrieval are recall and precision. *Precision* is the probability that a text presented as relevant to the user indeed is relevant. This metric is calculated as the number of relevant texts classified as relevant divided by the total number of texts classified as relevant. Assuming feedback (i.e. the specification of true class labels) for all texts presented to the user is available, it can always be evaluated. Even if precision can be used to detect changes under certain circumstances, it is likely to fail in detecting changes in case texts of relevant topics are classified as irrelevant because the precision does not take into account texts kept from the user as irrelevant. Since the detection of this type of change was found to be of major importance, we will not use precision other than for presenting empirical results. *Recall* is the probability that the filter lets through relevant texts to the user. It is estimated by the number of relevant texts classified as relevant divided by the total number of relevant texts. Thus it could be used to detect an increasing number of relevant texts kept back from the user. However, it can only be calculated when the true class labels of all texts are available. The *error rate* is the probability that a text is misclassified. It is calculated as the number of misclassified texts divided by the total number of texts. Thus, its calculation also requires the true class labels of all texts.

Instead of using these common performance measures, we derive an indicator from intermediate classification results without knowledge of the true class labels. We assume that a confidence score for each class can be obtained from the classifier. A change which would require adaptation is suspected when the current classification confidences drop significantly below the previously observed confidences. We will observe the confidence scores for the *irrelevant* class only since we focus on detecting relevant texts which were classified as irrelevant. We define a virtual reject class V which consists of all texts that will be classified as irrelevant with a confidence that is below a certain confidence threshold θ. The indicator ν_{reject} is defined as the number of texts in V divided by current batch size. When ν_{reject} increases significantly as determined by the Shewhart control chart, we conclude that the classification confidences have decreased and a change might have occurred.

The key problem of this approach is to define the confidence threshold θ. The objective is to find a value for θ such that most of the irrelevant training texts have scores above θ and most of the relevant training texts have scores below θ. Having performed k-fold cross-validation on the training texts as for getting unbiased confidence scores, we split the scores obtained for the irrelevant class into two sets. Let S_{rel} be the set of these scores for texts actually being relevant and S_{non} be the set of scores belonging to irrelevant texts. We heuristically express the term *most scores of irrelevant text being above θ* by the p_{non}–th percentile of S_{non} (with $p_{\text{non}} \leq 50$ since most scores should be above this value) and the term *most scores of relevant texts being below θ* by the p_{rel}–th percentile of S_{rel} (with $p_{\text{rel}} \geq 50$ since most scores should be below this value). These expressions are combined by a function f to obtain the threshold as

$$\theta = f(p_{\text{rel}}\text{-th percentile}(S_{\text{rel}}), p_{\text{non}}\text{-th percentile}(S_{\text{non}}))\,.$$

By setting the parameters p_{rel} and p_{non} and choosing an appropriate function f the importance of either excluding irrelevant texts from V or rather capturing more relevant texts can be stressed. In the following experiments, we will determine p_{rel} and p_{non} such that $p_{\text{rel}} = 100 - p_{\text{non}}$ and the corresponding percentiles approximately break even. We then evaluate θ as the mean of these percentiles. In fact, this approximation corresponds to using the point of intersection of the density functions of two overlapping distributions.

Note that the reject class is only virtual, i.e. the classifier is still forced to decide whether or not a text is relevant, for two reasons. First, determining a good threshold for the reject class with the objective to detect changes need not be optimal for deciding about the class labels since only the confidence scores for the irrelevant class are considered while ignoring the scores for the relevant class. Second, this makes it easier to compare the performance of this information filter to other approaches.

5 Adapting to Changes

An information filter must be adapted to changes in order to retain classification accuracy. Basically, there are two ways of adapting a classifier. On the one hand, the classifier can be completely relearned from scratch based only on a currently representative training set. On the other hand, an existing classifier can be updated based on some current examples.

The difficulty of the first approach is to provide a truly representative training set for the current situation, see [Nakhaeizadeh *et al.*, 1998] for example. Here, we simply assume that the texts of the most recent batch are always representative for the current situation.

Updating a classifier based on some new examples brings up the question of how to combine the knowledge inherent to the existing classifier with new examples in order to yield a better classifier. This is commonly done by incremental learning algorithms. Note, though, that usually the design goal in incremental learning is to produce a model that does not depend on the sequence in which the training examples are presented (e.g., see [Utgoff, 1989]). For our problem, however, this may not be the case. The possibilities of updating a classifier crucially depends on the selected learning algorithm including the preprocessing of texts.

In text categorization, approaches that model prototypes for representing different classes are very common. Often they are based on Rocchio's method for relevance feedback [Rocchio, 1971]. The class label of a new text is then determined based on the similarity between the prototypes and the new text. If the prototypes can be modeled individually, updating these classifiers in case of a change is straight forward if topics in the text stream can be associated with the available prototypes. Another well-known approach is the *Naïve Bayes Classifier* (see [Dumais *et al.*, 1998], for example). This approach is also very suitable for updating an existing classifier since the text and word frequencies required for calculating the prior probabilities can simply be increased as new examples arrive. Yet, there are other approaches for which updating an existing classifier is more difficult. An example for this category is the *Support Vector Machine (SVM)* which has become very popular recently (e.g., see [Joachims, 1998]). The SVM tries to find a good representation for the boundary between two classes. Therefore, changes in the class structure due to changing topics will have a rather global impact on the learned model and associating the change with parts of the classifier is much harder.

In this paper, the main focus is on detecting changes and therefore we will simply relearn our classifiers from scratch in case changes have been detected. Generally, however, updating an existing classifier should be preferred because there is more potential for reducing the amount of extra work necessary for providing new training examples.

6 Empirical Results

6.1 Experimental Setup

The experiments made for the evaluation of our adaptive information filters are based on a subset of the *Reuters–21578* collection (which is publicly available at *http://www.research.att.com/~lewis/reuters21578.html*) and a subset of the collection of Usenet articles from 20 different newsgroups as described in [Joachims, 1997].

A change scenario is simulated for each data set by randomly splitting the corresponding texts into 21 batches. The first batch serves as the initial training set while the remaining 20 batches represent the temporal development. The texts of each batch are classified as if they were new texts of an incoming stream. Based on these texts an indicator may be evaluated for the purpose of detecting changes. Finally these texts may be used for adapting the information filter.

For the task of classification, a simple similarity-based classifier is applied which is a variant of Rocchio's method for relevance feedback [Rocchio, 1971] and is described as the *Find Similar* method by [Dumais *et al.*, 1998]. The classifier models each of the two relevance classes with exactly one prototype. Each prototype is the average (or centroid) of all vector representation of texts of the corresponding class. For representing texts as vectors, some stop words (like ”and”, ”or”, etc.) and words that occurred fewer than five times in the training set are removed. From the remaining set of words, a maximum of 1000 words per relevance class is selected according to the mutual information measure described in [Dumais *et al.*, 1998].

For the evaluation of our adaptive approaches, the performance measures *recall* and *precision* as well as the *error rate* as defined in section 4 are used. However, since the objective is to minimize the required amount of extra cost in terms of user feedback, we also introduce the following two characteristics. Firstly, the amount of user feedback for change detection $\gamma_d \in [0, 1]$ which is calculated as the proportion of texts found irrelevant which the user must read nonetheless in order to provide true class labels. Hence, an approach that does not require user feedback at all yields $\gamma_d = 0$ while an approach that requires complete feedback for all texts found irrelevant yields $\gamma_d = 1$. Secondly, we determine the value μ which counts the number of detected changes and thus the number of adaptations. This value is proportional to the amount of user feedback necessary for providing new examples when adapting an information filter. It further shows whether an indicator is reliably detecting changes. Consequently, a good adaptive approach should have γ_d near 0 and μ close to the number of actual changes while the performance should be as good as that of approaches that would operate with complete user feedback.

In the following, results of four approaches averaged over 10 trials with different random seeds are presented for each data set. The first approach does not adapt to changes while the other three approaches are adaptive. Two of them are intractable because they require complete user feedback and serve as benchmarks only. The latter uses the change detection method explored in section 4.

6.2 The Reuters–21578 Collection

This text corpus consists of 21 578 news stories that have been assigned to one or more of a large number of categories (such as corporate acquisitions, earnings, money market, grain, and interest). Two of the largest categories, namely *corporate acquisitions* and *earnings*) are selected as relevant topics (dropping 19 stories which have assignments to both of these categories). The other categories are combined to represent the irrelevant class. Table 1 shows the resulting topics and the number of texts assigned to them. The change scenario, a *shift* between two relevant topics, is simulated within the 21 batches as follows. Up to batch 7, there are only relevant texts of topic *ACQ*, in batch 8 are texts of both relevant topics, and from batch 9 on, there are only texts of topic *EARN*. This yields a total of 285 relevant texts per batch. In each batch, there are 722 text of the irrel-

Topic	Description	Texts
ACQ	Corporate Acquisitions	2 429
EARN	Earnings	3 968
OTHER	everything else	15 162
	Total	21 559

Table 1: Selected topics of the Reuters–21578 collection.

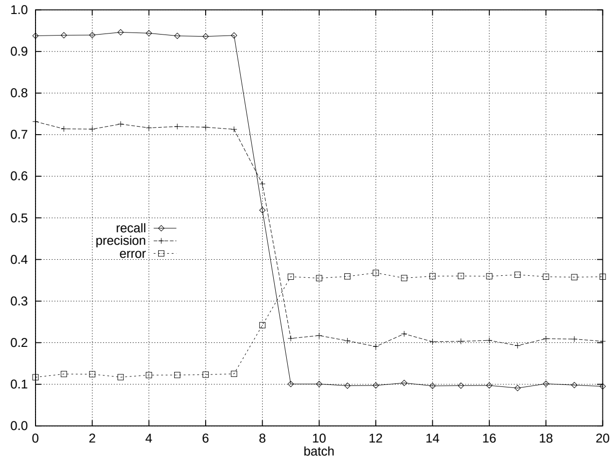

Figure 1: Approach without adaptation to changes.

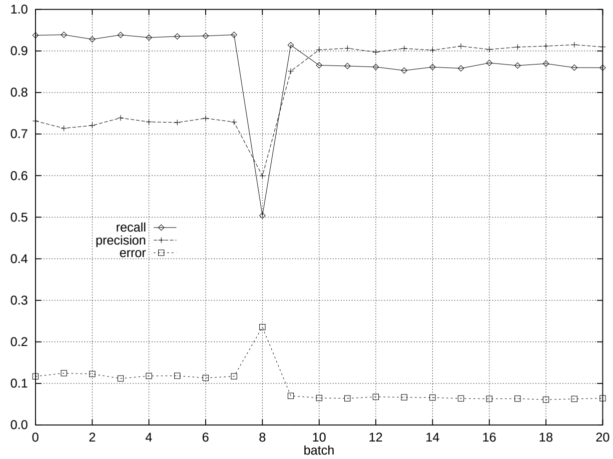

Figure 2: Approach that relearns after each batch.

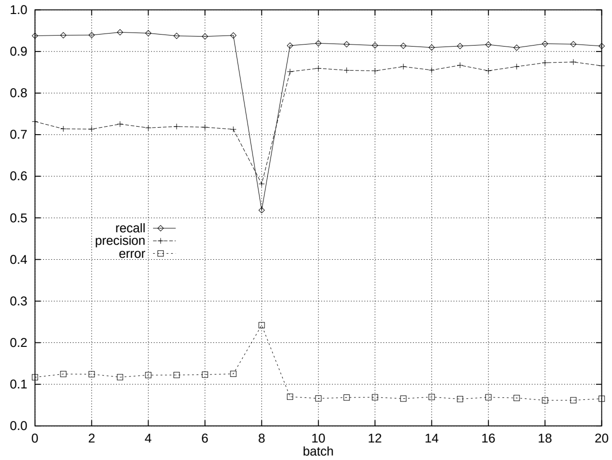

Figure 3: Approach that monitors error rates.

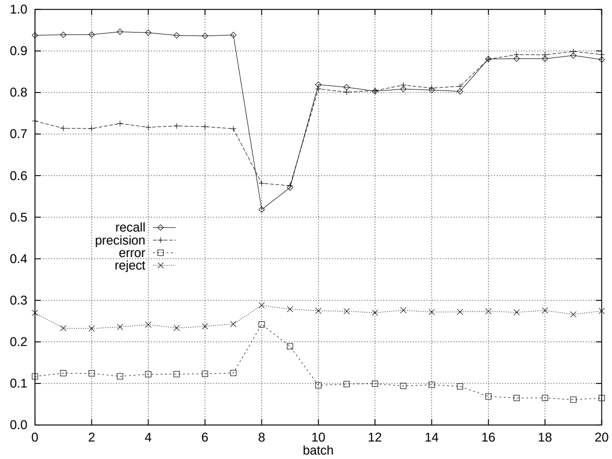

Figure 4: Approach that monitors confidences.

evant class *OTHER*, yielding a total of 1 007 texts per batch with 412 texts of the collection being discarded.

Figure 1 shows the performance of the approach that does not adapt to changes. The information filter is learned based on the training texts of the first batch and is then left unchanged ($\mu = 0$). Hence, there is no attempt to detect changes (thus $\gamma_d = 0$). Initially, the performance is quite good. After the change has occurred, however, the performance significantly decreases and is no longer acceptable.

As mentioned in the introduction, an alternative yet intractable approach for coping with changes, is to *relearn* the information filter in regular intervals no matter if there are any changes. While there is no attempt to detect changes ($\gamma_d = 0$), the required user feedback for providing new training data is prohibitive due to adapting after each batch ($\mu = 20$). Figure 2 shows the performance of this benchmark approach. The former level of performance can be regained after the change has occurred. In fact, the precision is even higher than before, indicating that the relevant *EARN* topic is easier to separate from the irrelevant class than the relevant *ACQ* topic. The results prove that the assumption that the texts of one batch are truly representative for learning the information filter holds for our experiments.

The other benchmark approach also operates with complete knowledge of the true class labels. Yet, instead of relearning after each batch, this knowledge is used to detect changes by monitoring the error rates (thus $\gamma_d = 1$). The change is detected immediately after it occurred in batch 8 as indicated by the peak of the curve labeled *error*. Hence the filter is adapted exactly once per trial ($\mu = 1$). The performance is about the same as that of the relearn approach, see figure 3. A slightly better recall is obtained with a slightly worse precision. Although not tractable, this benchmark shows that it is possible to detect shifts in the domain by monitoring an appropriate indicator with Shewhart control charts.

The only tractable adaptive approach monitors classification confidences and therefore requires no user feedback ($\gamma_d = 0$). Based on the result of 5-fold cross-validation and the break-even heuristic for the percentiles of the score distributions described in section 4, the threshold is set to about $\theta = 0.13$ averaged over all trials. Figure 4 shows that the performance is only slightly worse than that of the benchmark approaches. Except for one false alarm, i.e. a change is suspected when there is none, the filter is adapted once per trial (thus $\mu = 1.1$). However, the adaptations take place right after the change occurred in only seven of ten trials, twice with a delay of one and once with a delay of seven batches (although just barely failing to detect the change before).

Table 2 summarizes the results for all approaches, showing average values of recall, precision and error rates over all batches as well as the average amount of user feedback for change detection and the average number of adaptations performed in each trial. All adaptive approaches at least regain the initial level of performance and differ only slightly with respect to performance measures. With respect to user feedback, however, the approaches significantly differ from each other. Only the approach that monitors the confidences is applicable in practice.

Approach	Recall	Prec.	Error	γ_d	μ
No Adaption	41.36%	40.35%	27.08%	0	0.0
Relearn	87.27%	82.62%	9.20%	0	20.0
Error Rates	90.38%	79.68%	9.49%	1	1.0
Confidences	84.67%	77.43%	10.96%	0	1.1

Table 2: Average values over all batches (Reuters).

6.3 The 20-Newsgroups Collection

A subset of 19 newsgroups (omitting *misc.forsale*) is selected from this data set. The remaining newsgroups are combined to five broader topics as shown in table 3. Topics 1 (*recreation*) and 2 (*computing*) are chosen to be relevant while the other topics are irrelevant. The change scenario within the 21 batches is simulated as follows. Up to batch 8, there are only texts of the relevant topic 1 while there are only texts of the relevant topic 2 from batch 10 on, and batch 9 contains texts of both relevant topics. In each batch, there is the same composition of irrelevant texts. Hence, there are 420 relevant and 432 irrelevant texts in each batch yielding a total of 852 texts per batch with some of the texts being discarded.

The results for this data set are quite similar to that of the Reuters collection. The performance of the approach that does not adapt to changes is depicted in figure 5. After the change has occurred in batch 7, the initial performance can not be retained. However, the precision does not decrease as much as it does in the Reuters collection.

Topic	Description	Newsgroups	Texts
1	Recreation	*rec.**	4 000
2	Computing	*comp.**	5 000
3	Sciences	*sci.**	4 000
4	Politics	*talk.politics.**	3 000
5	Religion	*alt.atheism, *.religion.**	2 997
	Total		18 997

Table 3: Topics derived from the 20 Newsgroups corpus.

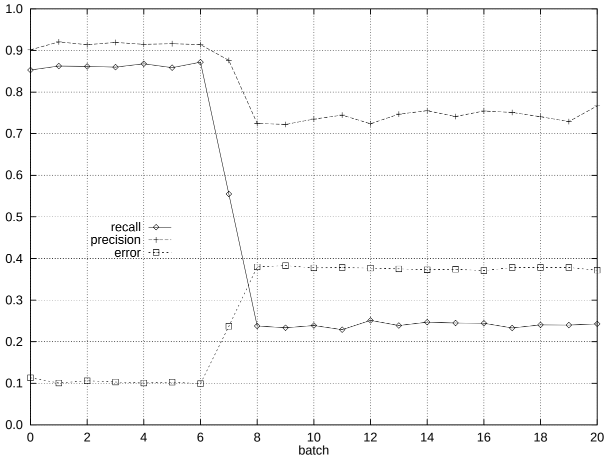

Figure 5: Approach without adaptation to changes.

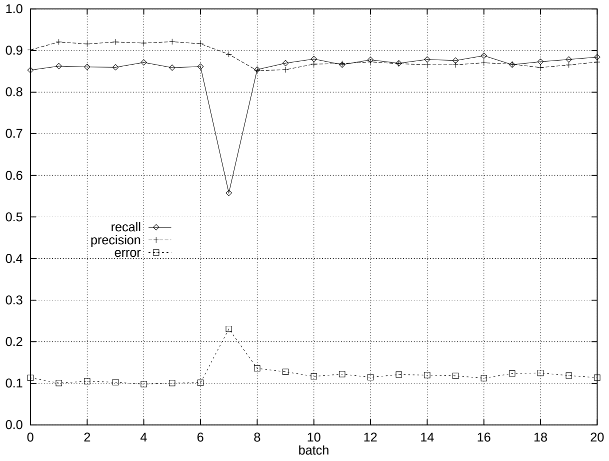

Figure 6: Approach that relearns after each batch.

Figure 6 and 7 show the results of the benchmark approaches operating with complete user feedback. Again, both approaches recover from the poor performance after the change has occurred and achieve about the same performance. The results of the relearn approach show that each batch is truly representative for completely learning the information filter from scratch. The peak of the *error* curve in figure 7 indicates where the adaptations for the approach monitoring the error rates take place. In each trial, the filter is adapted once right after the change has occurred in batch 7, and there is a false alarm in one of the trials. Hence, we obtain an average of $\mu = 1.1$ adaptations per trial.

The results of the only tractable adaptive approach (monitoring confidences) is shown in figure 8. Here, the threshold is set to about $\theta = 0.09$ averaged over all trials after performing 5-fold cross-validation and applying the break-even heuristic for the percentiles of the score distributions. The performance is about as good as that

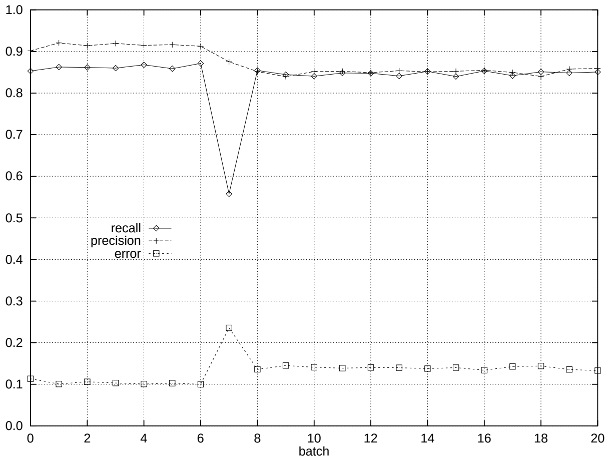

Figure 7: Approach that monitors error rates.

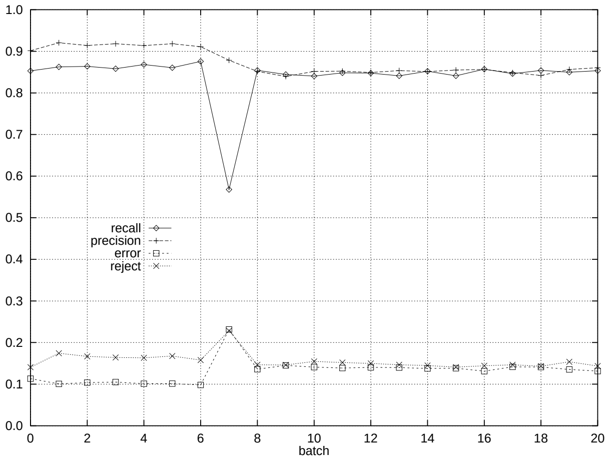

Figure 8: Approach that monitors confidences.

of the benchmark approaches. In each trial the change is detected as soon as it occurred in batch 7. This is indicated by the peak of the *reject* curve. However, in all trials there is a total of 9 false alarms, yielding $\mu = 1.9$. Although this increases the amount of user feedback required for providing new training texts, all changes are reliably detected.

Average values of recall, precision and error rates over all batches as well as the average amount of user feedback for change detection and the average number of adaptations performed in each trial are given in table 4. As for the Reuters collection, all adaptive approaches regain the initial level of performance and differ only slightly with respect to the performance measures. With respect to user feedback, however, the approaches significantly differ from each other. In fact, the adaptive approach monitoring the confidences is still the only practically applicable approach despite the false alarms.

Approach	Recall	Prec.	Error	γ_d	μ
No Adaption	44.29%	80.05%	28.72%	0	0.0
Relearn	85.46%	88.25%	12.04%	0	20.0
Error Rates	83.76%	87.18%	13.28%	1	1.1
Confidences	83.93%	87.21%	10.83%	0	1.9

Table 4: Average results over all batches (Newsgroups).

7 Conclusions and Future Work

Our experiments show that detecting changes is possible without user feedback. Although there are some false alarms in the change detection process of the approach monitoring the classification confidences, all changes but one are reliably detected. Thus, the amount of extra work required to be done by the user in order to retain the desired accuracy in classification can be significantly decreased.

The change scenarios examined so far exhibit only a rather drastic change in the domain. In real-world applications, however, changes usually are much slower and less radical and therefore more difficult to detect. It is a well-known weakness of the Shewhart control chart that small changes may not be detected. Therefore, *CUSUM control charts* could be used in addition (see [Montgomery, 1997]). This technique accumulates deviations from an expected value and can thus detect small changes which occur successively while the Shewhart control chart test whether only a single observation of an indicator is within acceptable variation. In future research, further evaluations with more realistic change scenarios will follow.

A major problem of our approach is that even quite radical changes need not have a discernible effect on the confidence scores. Further experiments show, for example, that the scores of new texts may be similar to those of texts of existing topics of the irrelevant class even if they belong to previously unseen topics. This problem especially occurs when the irrelevant class consists of many diverse topics. An idea for coping with this problem is to model each class with several rather than one prototype. Here, text clustering may be useful for determining topics inherent to each relevance class. Furthermore, in future research indicators based on text properties will also be applied for detecting changes

So far only an adaptation strategy that relearns an information filter from scratch was applied. However, approaches that update existing filters based on some new training data should also be considered because they provide more potential for further reducing the amount of required user feedback. In this context, text clustering may also be used in order to group new texts helping the user to label these texts. Furthermore, the idea of *active learning* as applied to text classification in [Lewis and Gale, 1994] should be considered.

References

[Allan *et al.*, 1998] J. Allan, J. Carbonell, G. Doddington, J Yamron, and Y. Yang. Topic detection and tracking pilot study final report. In *Proceedings of* *the DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[Dumais *et al.*, 1998] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representation for text categorization. In *Proceedings of the Seventh International Conference on Information and Knowledge Management*, 1998.

[Joachims, 1997] T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *International Conference on Machine Learning*, 1997.

[Joachims, 1998] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML)*. Springer Verlag, 1998.

[Klinkenberg and Renz, 1998] R. Klinkenberg and I. Renz. Adaptive information filtering: Learning in the presence of concept drifts. In *Learning for Text Categorization*, pages 33–40, Menlo Park, California, 1998. AAAI Press.

[Lanquillon, 1998] C. Lanquillon. Dynamic aspects in neural classification. In G. Nakhaeizadeh and E. Steurer, editors, *Application of Machine Learning and Data Mining in Finance, ECML'98 Workshop Notes*, Chemnitz, Germany, 1998. Chemnitzer Informatik-Berichte CSR-98-06.

[Lewis and Gale, 1994] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 3–12, London, 1994. Springer-Verlag.

[Lewis, 1995] David D. Lewis. Evaluating an optimizing autonomous text classification systems. In *Proceedings of the Eighteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*, pages 246–254, July 1995.

[Montgomery, 1997] D.C. Montgomery. *Introduction to Statistical Quality Control*. Wiley, New York, 3rd edition, 1997.

[Nakhaeizadeh *et al.*, 1998] G. Nakhaeizadeh, C. Taylor, and C. Lanquillon. Evaluating usefulness for dynamic classification. In *Proceedings of The Fourth International Conference on Knowledge Discovery & Data Mining*, pages 87–93, New York, 1998.

[Rocchio, 1971] J.J. Jr. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.

[Utgoff, 1989] P. E. Utgoff. Incremental learning of decision trees. *Machine Learning*, 4:161–186, 1989.