Adaptive Information Filtering: Learning Drifting Concepts

Ralf Klinkenberg*and Ingrid Renz**

* Artificial Intelligence Unit, Computer Science Department, University of Dortmund, Baroper Str. 301, D-44221 Dortmund, Germany, E-Mail: klinkenberg@ls8.cs.uni-dortmund.de

** Research and Technology, Daimler-Benz AG, P. O. Box 2360, D-89013 Ulm, Germany, E-Mail: renz@dbag.ulm.daimlerbenz.com

Abstract. The task of information filtering is to classify texts from a stream of documents into relevant and non-relevant, respectively, with respect to a particular category or user interest, which may change over time. A filtering system should be able to adapt to such concept changes. This paper explores methods to recognize concept changes and to maintain windows on the training data, whose size is either fixed or automatically adapted to the current extent of concept change. Experiments with two simulated concept drift scenarios based on real-world text data and eight learning methods are performed to evaluate three indicators for concept changes and to compare approaches with fixed and adjustable window sizes, respectively, to each other and to learning on all previously seen examples. Even using only a simple window on the data already improves the performance of the classifiers significantly as compared to learning on all examples. For most of the classifiers, the window adjustments lead to a further increase in performance compared to windows of fixed size. The chosen indicators allow to reliably recognize concept changes.

Keywords. Machine Learning, Adaptive Information Filtering, Text Classification, Concept Drift

1 Introduction

With the amount of online information and communication growing rapidly, there is an increasing need for automatic information filtering. Information filtering techniques are used, for example, to build personalized news filters, which learn about the news-reading preferences of a user, or to filter e-mail. The concept underlying the classification of the texts into relevant and non-relevant may change. Machine learning techniques ease the adaption to (changing) user interests.

This paper focuses on the aspect of changing concepts in information filtering. After reviewing the standard feature vector representation of text and giving some references to other work on adaption to changing concepts, this paper describes indicators for recognizing concept changes and uses some of them as a basis for a window adjustment heuristic that adapts the size of a time window on the training data to the current extent of concept change. The indicators and data management approaches with windows of fixed and adaptive size are evaluated in two simulated concept drift scenarios on real-world text data.

2 Text Representation

In Information Retrieval, words are the most common representation units for text documents and it is usually assumed, that their ordering in a document is of minor importance for many tasks. This leads to an attribute-value representation of text, where each distinct word w_i corresponds to a feature with the number of times it occurs in the document d as its value (*term frequency*, $TF(w_i, d)$). The length of the feature vector is reduced by considering only words as features that occur at least 3 times in the training data and are not in a given list of stop words (like “the”, “a”, “and”, etc.).

For some of the learning methods used in the experiments described in this paper, a subset of the features is selected using the *information gain* criterion [11], to improve the performance of the learner and/or speed up the learning process. The remaining components w_i of the document feature vector are then weighted by multiplying them with their *inverse document frequency (IDF)*. Given the *document frequency* $DF(w_i)$, i. e. the number of documents word w_i occurs in, and the total number of documents $|D|$, the inverse document frequency of word w_i is computed as $IDF(w_i) = \log \frac{|D|}{DF(w_i)}$. Afterwards each document feature vectors is normalized to unit length to abstract from different document lengths.

In the experiments described in this paper, the performance of a classifier is measured by the three metrics accuracy, recall, and precision. *Accuracy* is the probability, that a random document is classified correctly. It is estimated as the number of correct classifications divided by the total number of classifications. *Recall* is the probability, that the classifier recognizes a relevant document as relevant, and is computed as the number of relevant documents classified as relevant divided by the total number of relevant documents. *Precision* is the probability, that a document classified as relevant actually is relevant. It is estimated by the number of relevant documents classified as relevant divided by the total number of documents classified as relevant. The metrics can be computed from a contingency table:

	Relevant	Non-relevant
Classified as relevant	a	b
Classified as non-relevant	c	d

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} \quad (1)$$

$$\text{Recall} = \frac{a}{a+c} \quad (2)$$

$$\text{Precision} = \frac{a}{a+b} \quad (3)$$

3 Adapting to Changing Concepts

In machine learning, changing concepts are often handled by time windows of fixed or adaptive size on the training data (see for example [15], [8]) or by weighting data or parts of the hypothesis according to their age and/or utility for the classification task ([6], [13]). The latter approach of weighting examples has already been used in information filtering by the incremental relevance feedback approach [1] and by [2]. In this paper, the earlier approach maintaining a window of adaptive size and explicitly recognizing concept changes is explored in the context of information filtering. More detailed descriptions of the methods described above and further approaches can be found in [5].

For windows of fixed size, the choice of a *"good" window size* is a compromise between fast adaptability (small window) and good and stable learning results in phases without concept change (large window). The basic idea of the *adaptive window management* is to adjust the window size to the current extent of concept drift. In case of a suspected concept drift or shift, the window size is decreased by dropping the oldest, no longer representative training instances. In phases with a stable concept, the window size is increased to provide a large training set as basis for good generalizations and stable learning results. Obviously, reliable indicators for recognizing concept changes play a central role in such an adaptive window management.

3.1 Indicators for Concept Drifts

Different types of indicators can be monitored to detect concept changes:

- *Performance measures* (e. g. the accuracy of the current classifier): independent of the hypothesis language, generally applicable.
- *Properties of the classification model* (e. g. the complexity of the current rules): dependent on a particular hypothesis language, applicable only to some classifiers.
- *Properties of the data* (e. g. class distribution, attribute value distribution, current top attributes according to a feature ranking criterion, or current characteristic of relevant documents like cluster memberships): independent of the hypothesis language, generally applicable.

The indicators of the window adjustment heuristic of the FLORA algorithms [15], for example, are the accuracy and the coverage of the current concept description, i. e. the number of positive instances covered by the current hypothesis divided by the number of literals in this hypothesis. Obviously the coverage can only be computed for rule-based classifiers. The SIFTER information filtering system [7] determines clusters of similar documents and monitors the probability to be relevant for documents of each cluster seperately. Changes in the user interest are recognized by changes of these probabilities. The performance of the classifier is not monitored. This approach is independent of the learning method underlying the system.

The window adjustment approach for text classification problems proposed in this paper only uses performance measures as indicators, because they can be applied across different learning methods and are expected to be the most reliable indicators. The computation of performance measures like accuracy requires user feedback about the true class of filtered documents. In some applications only partial user feedback is available to the filtering system. For the experiments described in this paper, complete feedback about all filtered documents is assumed. In most information filtering tasks, the irrelevant documents significantly outnumber the relevant documents and a default rule predicting all new documents to be irrelevant achieves a high accuracy, because accuracy does not distinguish between different types of misclassifications. As equation (1) shows, the accuracy does not only depend on a, but on d, the number of non-relevant documents classified correctly, as well. If d is assumed to be a very large, constant number, the accuracy does not reflect concept changes as much as recall and precision (equations (2) and (3)) and hence is alone only of limited use as performance metric and indicator for text classifiers. Therefore the metrics recall and precision are used as indicators in addition to accuracy, because they assess the performance on the smaller, usually more important class of relevant documents.

3.2 Adaptive Window Adjustment

The texts are presented to the filtering system in batches. Each batch is a sequence of several texts from the stream of documents to be filtered. In order to recognize concept changes, the values of the three indicators accuracy, recall, and precision are monitored over time and the average value and the standard sample error are computed for each of these indicators based on the last M batches at each time step. Each indicator value is compared to a confidence interval of α times the standard error around the average value of this indicator. The confidence niveau α is a user-defined constant ($\alpha > 0$). If the indicator value is smaller than the lower end point of this interval, a concept change is suspected. In this case, a further test determines, whether the change is abrupt and radical (*concept shift*) or rather gradual and slow (*concept drift*). If the current indicator value is smaller than its predecessor times a user-defined constant β ($0 < \beta < 1$), a concept shift is suspected, otherwise a concept drift.

In case of a concept shift, the window is reduced to its minimal size, the size of one batch ($|B|$), in order to drop the no longer representative old examples as fast as possible. If only a concept drift has been recognized, the window is reduced less radically by a user-defined reduction rate γ ($0 < \gamma < 1$). Thereby some of the old, still at least partially representative data for the current concept is kept. This establishes a compromise between fast adaptivity via a reduction of the window size and stable learning results as a result of a large training data set. If neither a concept shift nor a drift is suspected, all seen examples are stored, in order to provide a training set of maximal size, because in case of a stable concept, text classifiers usually perform the better, the more training examples they have.

While in real applications an upper bound for the size of the adaptive window seems reasonable, no such bound was used for the experiments described in this paper. Figure 1 describes the window adjustment heuristic. For the first M_0 initial batches, the window size is not adapted, but left at its initial value of $|W_0|$ to establish the average indicator values and their standard errors. $|W_t|$ denotes the current window size and $|W_{t+1}|$ the new window size. $|B|$ is the number of documents in a batch. Acc_t is the current accuracy value, Acc_{t-1} is the previous accuracy value, $Avg_M(Acc)$ is the average accuracy of the last M batches, and $StdErr_M(Acc)$ is the standard error of the accuracy on the last M batches. Rec_t, Rec_{t-1}, $Avg_M(Rec)$, and $StdErr_M(Rec)$ denote the corresponding recall values, and $Prec_t$, $Prec_{t-1}$, $Avg_M(Prec)$, and $StdErr_M(Prec)$ the corresponding precision values.

4 Experiments

The experiments use a subset of the data set of the *Text REtrieval Conference (TREC)* consisting of English business news texts. Each text is assigned to one or several categories. Table 1 shows the names and sizes of the categories 1, 3, 4, 5, and 6 used here. For the experiments, two concept change scenarios are simulated. The texts are randomly split into 20 batches of equal size containing 130 documents each[1]. The texts of each category are distributed as equally as possible to the 20 batches.In the first scenario (*scenario A*), first documents of category 1 (Antitrust Cases Pending) are considered relevant for the user interest and all other documents irrelevant. This changes abruptly (concept shift) in batch 10, where documents of category 3 (Joint Ventures) are relevant and all others irrelevant. Table 2 specifies the probability of being relevant for documents of category 1 and 3 for each time step (batch). Classes 4, 5, and 6 are never relevant. In the second scenario (*scenario B*), again first documents of category 1 (Antitrust Cases Pending) are considered relevant for the user interest and all other documents irrelevant. This changes slowly (concept drift) from batch 8 to batch 12, where documents of category 3 (Joint Ventures) are relevant and all others irrelevant. Table 3 specifies the probability of being relevant for documents of category 1 and 3 for each time step (batch). Classes 4, 5, and 6 are never relevant.

Category	Name of the Category	Number of Documents
1	Antitrust Cases Pending	400
3	Joint Ventures	842
4	Debt Rescheduling	355
5	Dumping Charges	483
6	Third World Debt Relief	528
	Total	2608

Table 1 Categories of the TREC data set used in the experiments.

4.1 Experimental Setup

The experiments are performed according to the batch learning scenario, i. e. the learning methods learn a new classification model whenever they receive a new batch of training documents. Each of the following *data management approaches* is tested in combination with each of the learning methods listed further below:

- *"Full Memory"*: The learner generates its classification model from all previously seen examples, i.e. it cannot "forget" old examples.
- *"No Memory"*: The learner always induces its hypothesis only from the least recently seen batch. This corresponds to using a window of the fixed size of one batch.
- Window of *"Fixed Size"*: A window of the fixed size of three batches is used.
- Window of *"Adaptive Size"*: The window adjustment heuristic (figure 1) is used to adapt the window size to the current concept drift situation.

[1]Hence, in each trial, out of the 2608 documents 8 randomly selected texts are not considered.

Procedure DetermineNewWindowSize $(|W_t|,\ M,\ \alpha,\ \beta,\ \gamma)$

if $((Acc_t < Avg_M(Acc) - \alpha \cdot StdErr_M(Acc))$ **and** $(Acc_t < \beta \cdot Acc_{t-1}))$ **or**
$((Rec_t < Avg_M(Rec) - \alpha \cdot StdErr_M(Rec))$ **and** $(Rec_t < \beta \cdot Rec_{t-1}))$ **or**
$((Prec_t < Avg_M(Prec) - \alpha \cdot StdErr_M(Prec))$ **and** $(Prec_t < \beta \cdot Prec_{t-1}))$

then $|W_{t+1}| := |B|$; /* concept shift suspected: reduce window size to one batch */

else if $(Acc_t < Avg_M(Acc) - \alpha \cdot StdErr_M(Acc))$ **or**
$(Rec_t < Avg_M(Rec) - \alpha \cdot StdErr_M(Rec))$ **or**
$(Rec_t < Avg_M(Prec) - \alpha \cdot StdErr_M(Prec))$

then $|W_{t+1}| := \max(|B|,\ |W_t| - \gamma \cdot |W_t|)$; /* concept drift suspected: reduce window size by $\gamma \cdot 100\%$ */

else $|W_{t+1}| := |W_t| + |B|$; /* stable concept suspected: grow window by one batch */

return $|W_{t+1}|$;

Figure 1 Window adjustment heuristic for text categorization problems.

Category	Relevance of the categories for each batch																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 2 Relevance of the categories in concept change scenario A (abrupt concept shift in batch 10).

For the adaptive window management approach, the initial window size is set to three batches ($|W_0| := 3 \cdot |B|$), the number of initial batches to five ($M_0 := 5$), and the number of batches for the averaging process to 10 ($M := 10$). The width of the confidence interval is set to $\alpha := 5.0$, the factor $\beta := 0.5$, and the window reduction rate $\gamma := 0.5$. These values are arbitrarily set and not result of an optimization.

The parameters of the *learning methods* listed below were found to perform well in a prelimenary experiment for a different classification task on the TREC data set, but are not optimized for the concept drift scenarios considered here: the *Rocchio Algorithm* [12] with $\alpha_{Rocchio} := 1.0$, $\beta_{Rocchio} := 1.0$, and a threshold θ determined via v-fold crossvalidation ($v = 4$), a *Naive Bayes Classifier* [4], the *PrTFIDF Algorithm* [4], a distance-weighted *k-Nearest Neighbors (k-NN)* method [10] with $k := 5$, the *Winnow Algorithm* [9] with learning rate $\gamma := 1.1$ and 40 iterations for learning, a *Support Vector Machine (SVM)* [14] with polynomial kernel and polynom degree one (= linear kernel), the symbolic rule learner *CN2* [3] with the default paramters for unordered rules, and the symbolic decision tree and rule learning system *C4.5* [11] with the default parameters to induce a decision tree, transform it to an ordered rule set, and post-prune the resulting rules. In the experiments described here, Winnow is not used as an online learner, but generates a new classification model on the current training set, whenever a new batch of documents is received. For Winnow, C4.5, and CN2 the 1000 best attributes according to the information gain criterion are selected. All other methods use all attributes. The results reported in the following sections are averaged over four trials

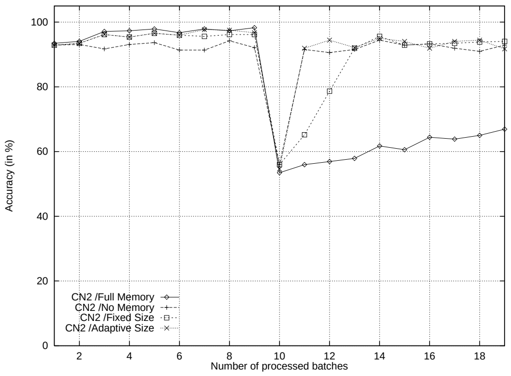

Figure 2 Accuracy of CN2 with the different data management approaches for scenario A.

for each combination of learning method, data management approach, and concept drift scenario.

4.2 Results for Scenario A (Concept Shift)

Table 4 shows accuracy, recall, and precision of all combinations of learning methods and data management approaches averaged over 4 trials according to scenario A (table 2). In addition, table 4 compares a pair of data management approaches in each of its three right most columns. Column "(2) vs. (1)" is the performance gain obtained by approach (2) (No Memory) over (1) (Full Memory). Accordingly, the last two columns compare the Adaptive Size approach to the approaches with fixed window size, i. e. No Memory and Fixed Size, respectively.

Column "(2) vs. (1)" shows that for all learning methods a significant improvement is achieved by using the simple No Memory window approach instead of learning on all known examples (Full Memory).

Category	Relevance of the categories for each batch																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.6	0.8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 3 Relevance of the categories in concept change scenario B (slow concept drift from batch 8 to batch 12).

		Full Memory (1)	No Memory (2)	Fixed Size (3)	Adaptive Size (4)	(2) vs. (1)	(4) vs. (2)	(4) vs. (3)
Rocchio	Accuracy	75.95%	84.63%	87.93%	89.38%	+08.68%	+04.75%	+01.45%
	Recall	49.77%	93.59%	87.41%	91.74%	+43.82%	–01.85%	+04.33%
	Precision	48.64%	61.77%	70.43%	72.91%	+13.13%	+11.14%	+02.48%
Naive Bayes	Accuracy	81.96%	93.59%	91.97%	93.97%	+11.63%	+00.38%	+02.00%
	Recall	68.51%	86.96%	84.67%	88.18%	+18.45%	+01.22%	+03.51%
	Precision	67.63%	87.32%	84.69%	87.68%	+19.69%	+00.36%	+02.99%
PrTFIDF	Accuracy	80.67%	88.18%	87.44%	88.87%	+07.51%	+00.69%	+01.43%
	Recall	85.33%	93.49%	93.31%	94.19%	+08.16%	+00.70%	+00.88%
	Precision	56.78%	67.66%	66.21%	64.42%	+10.88%	–03.24%	–01.79%
k-NN	Accuracy	79.32%	91.26%	90.14%	92.33%	+11.94%	+01.07%	+02.19%
	Recall	49.82%	76.60%	74.45%	80.74%	+26.78%	+04.14%	+06.29%
	Precision	63.34%	87.14%	84.29%	87.02%	+23.80%	–00.12%	+02.73%
Winnow	Accuracy	74.48%	89.94%	89.15%	91.64%	+15.46%	+01.70%	+02.49%
	Recall	41.44%	70.09%	70.77%	78.33%	+28.65%	+08.24%	+07.56%
	Precision	48.46%	83.12%	82.03%	85.95%	+34.66%	+02.83%	+03.92%
SVM	Accuracy	79.48%	92.64%	91.80%	94.48%	+13.16%	+01.84%	+02.68%
	Recall	51.03%	74.24%	77.11%	83.95%	+23.21%	+09.71%	+06.84%
	Precision	64.65%	91.27%	87.32%	91.49%	+26.62%	+00.22%	+04.17%
CN2	Accuracy	77.72%	90.50%	90.16%	92.45%	+12.78%	+01.95%	+02.29%
	Recall	41.20%	68.45%	69.68%	76.74%	+27.25%	+08.29%	+07.06%
	Precision	56.49%	85.89%	85.37%	89.56%	+29.40%	+03.67%	+04.19%
C4.5	Accuracy	78.49%	91.40%	90.29%	92.83%	+12.91%	+01.43%	+02.54%
	Recall	49.22%	79.02%	76.49%	83.47%	+29.80%	+04.45%	+06.98%
	Precision	51.24%	82.10%	81.84%	86.03%	+30.86%	+03.93%	+04.19%
Average	Accuracy					+11.76%	+01.73%	+02.13%
	Recall					+25.76%	+04.36%	+05.43%
	Precision					+23.63%	+02.35%	+02.86%

Table 4 Accuracy, recall and precision of all learning methods combined with all data management approaches for scenario A averaged over 4 trials with 20 batches each.

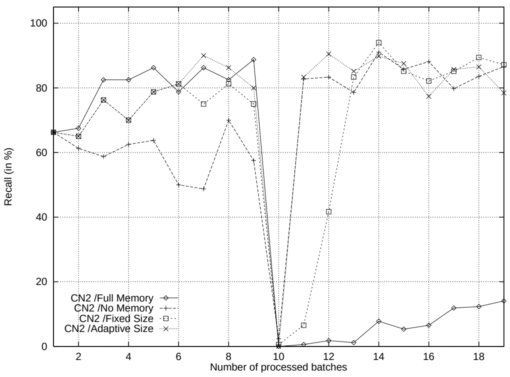

Figure 3 Recall of CN2 with the different data management approaches for scenario A.

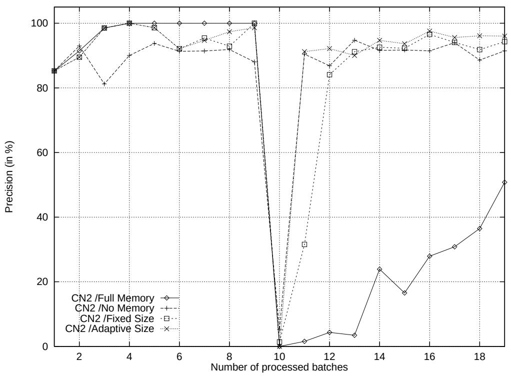

Figure 4 Precision of CN2 with the different data management approaches for scenario A.

The average gain is 11.8% in accuracy, 25.8% in recall, and 23.6% in precision. An additional improvement is achieved by using the Adpative Size approach instead of an approach with fixed window size (see columns "(4) vs. (2)" and "(4) vs. (3)" in table 4). The average gain of Adaptive Size compared to the best approach with a window of fixed size is 1.7% in accuracy, 4.4% in recall, and 2.4% in precision. A closer look at the last two columns of table 4 shows, that some methods like CN2, C4.5, the SVM, Win-

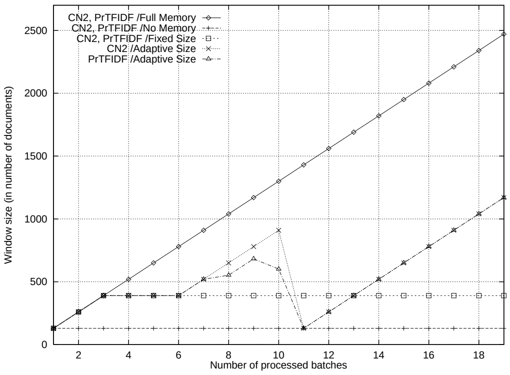

Figure 5 Window size for CN2 and PrTFIDF with the different data management approaches for scenario A.

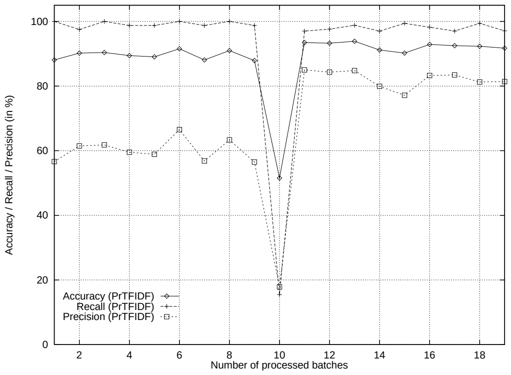

Figure 6 Accuracy, recall, and precision for PrTFIDF with Adaptive Size for scenario A.

now, Rocchio, and k-NN gain significantly by using Adaptive Size, while other methods like PrTFIDF and Naive Bayes do not show a significant improvement. For PrTFIDF, the precision actually drops by more than 3.2%. Figures 2 to 5 show the values of the three indicators and the window size over time for the learning method CN2 in combination with all data management approaches and thereby allow a more detailed analysis of the results than table 4. The figures 2 to 4 with the accuracy, recall, and precision values of CN2 show two things. First, in this scenario all three indicators can be used to easily detect the concept shift, because their values decrease very significantly in the batch the shift occurs in (batch 10). Recall and precision indicate the shift even more clearly than accuracy.

Second, in this scenario the data management approaches demonstrate their typical behaviour in relation to each other. Before the shift, the Full Memory approach has the advantage of the largest training set and hence shows the most stable performance and outperforms the other three approaches, but recovers only very slowly from its break-down after the concept shift. The Fixed Size approach shows a relatively good performance in phases with stable target concept, but needs several batches to recover after the concept shift. The No Memory approach offers the maximum flexibility and recovers from the shift after only one batch, but in phases with a stable concept, this approach is less stable and performs worse than the other approaches. In this scenario and in combination with CN2, the Adaptive Size approach obviously manages to show a high and stable performance in stable concept phases *and* to adapt very fast to the concept shift. Hence Adaptive Size here is able to combine the advantages of different window sizes.

Figure 5 shows the window size of the four data management approaches in combination with CN2 over time. The window of the Full Memory approach grows linearly in the number of examples seen, while the No Memory approach always keeps a window of one batch size. The window of the Fixed Size approach grows up to a size of three batches, which it keeps afterwards. The Adaptive Size window grows up to its initial size of three batches (user-defined constant $|W_0|$) and keeps this size until the last of the initial batches for establishing the average values and standard errors (user-defined constant $M_0 = 5$). From the sixth batch on, the window adjustment becomes active and the window grows until the concept shift occurs in batch 10. Then the window is set to its minimal size of one batch, but starts growing again immediately afterwards, because no further shift or drift is detected.

Figure 6 with the indicator values for PrTFIDF in combination with the adaptive window management shows, that the three indicators work for PrTFIDF as well, i. e. they indicate the concept shift by a significant decrease in their values. Figure 5 with the window size for PrTFIDF in combination with the four data management approaches shows that the window adjustment for PrTFIDF works almost as the one for CN2. But, unlike for CN2, this is not reflected by an increase in the performance of PrTFIDF. A window of the size of one batch already seems to be sufficiently large for PrTFIDF in this scenario, so that the window adjustments cannot provide any improvement.

4.3 Results for Scenario B (Concept Drift)

Table 5 compares accuracy, recall, and precision of the same pairs of data management approaches as table 4 in combination with all learning methods averaged over 4 trials according to scenario B (table 3). Like in scenario A, using the simple No Memory approach instead of the Full Memory approach yields significant performance improvements (column "(2) vs. (1)"). On average, accuracy is improved by 10.2%, recall by 22.5%, and precision by 21.2%. The average increase in performance gained by the Adaptive Size approach over the best approach with fixed window size is 0.3% in accuracy, 1.3% in recall, and 0.9% in precision. (columns "(4) vs. (2)" and "(4) vs. (3)" in table 5). The average positive effect of the window adjustments

		(2) vs. (1)	(4) vs. (2)	(4) vs. (3)
Rocchio	Accuracy	+08.16%	+04.79%	+01.53%
	Recall	+41.63%	–03.15%	+02.77%
	Precision	+11.12%	+11.91%	+03.31%
Naive Bayes	Accuracy	+09.85%	+00.28%	+00.44%
	Recall	+16.15%	+01.94%	+01.93%
	Precision	+17.71%	–00.39%	+00.83%
PrTFIDF	Accuracy	+06.17%	–00.58%	–00.39%
	Recall	+09.86%	+00.70%	–01.80%
	Precision	+08.75%	–00.94%	–00.70%
k-NN	Accuracy	+11.41%	–00.59%	–00.41%
	Recall	+24.09%	+00.14%	–00.10%
	Precision	+23.88%	–01.30%	+00.23%
Winnow	Accuracy	+12.66%	+00.15%	–00.14%
	Recall	+23.14%	+04.48%	+01.10%
	Precision	+29.42%	+00.52%	+00.50%
SVM	Accuracy	+12.15%	+00.83%	+00.54%
	Recall	+17.27%	+07.52%	+01.58%
	Precision	+23.74%	–00.73%	+01.49%
CN2	Accuracy	+10.78%	+01.65%	+00.47%
	Recall	+22.97%	+03.51%	+02.36%
	Precision	+25.86%	+03.60%	+00.25%
C4.5	Accuracy	+10.21%	+01.12%	+00.68%
	Recall	+24.49%	+02.99%	+02.89%
	Precision	+28.89%	+02.80%	+01.03%
Average	Accuracy	+10.17%	+00.96%	+00.34%
	Recall	+22.51%	+02.26%	+01.34%
	Precision	+21.17%	+01.93%	+00.87%

Table 5 Accuracy, recall and precision of all learning methods for scenario B compared for the data management approaches Full Memory (1), No Memory (2), Fixed (3) and Adaptive Size (3).

is obviously smaller than in scenario A. While three methods show no significant positive or even a negative effect through the adjustments, namely PrTFIDF, k-NN, Bayes, most of the methods, i. e. CN2, C4.5, Rocchio, Winnow, and the SVM, profit by the window adjustments. As figure 7 shows for the example CN2, the three indicators work reliably in scenario B. Recall and precision again indicate the concept change much better than accuracy. The window size of Adaptive Size with CN2 (figure 8) shows, that the window adjustment works in this scenario as well. The concept drift is already detected in batch 9 and the window size is reduced accordingly. The reduction of the window size continues until the end of the concept drift in batch 12. The fact, that the window was not radically set to its minimal size of one batch, shows, that the concept drift was not mistakenly suspected to be a concept shift. Although PrTFIDF does not profit by the window adjustments as CN2, its window adjustment works almost as well as for CN2 (figure 8).

4.4 Setting the Parameters of the Window Adjustment Heuristic

The parameters α, β, and γ of the window adjustment heuristic were fixed in the experiments described in

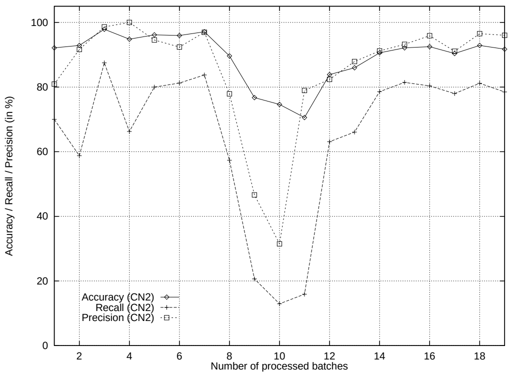

Figure 7 Accuracy, recall, and precision for CN2 with Adaptive Size for scenario B.

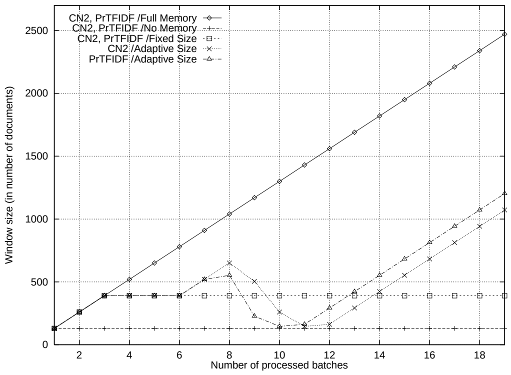

Figure 8 Window size for CN2 and PrTFIDF with the different data management approaches for scenario B.

the two previous sections. In order to evaluate how much the performance of the classifiers depends on the choice of the values for these parameters, an additional experiment is performed on scenario B, whose concept drift is a little bit more difficult to recognize than the concept shift of scenario A. Table 6 shows the results of applying the learning methods PrTFIDF and C4.5 with all combinations of $\alpha \in \{2.5, 5.0, 7.5\}$, $\beta \in \{0.25, 0.50, 0.75\}$, and $\gamma \in \{0.25, 0.50, 0.75\}$. For both, PrTFIDF and C4.5, the choice of a good value for α seems to be more crucial than the choices of β and γ. If α, which describes the width of the confidence interval for admissible drops in performance, is too large, the concept drift of scenario B is no longer properly recognized and the performance of the classifiers drops significantly. Otherwise the window adjustment heuristic seems to be fairly robust to the choice of the parameters α, β, and γ.

5 Conclusions

This paper describes indicators for recognizing concept changes and uses some of them as a basis for a window adjustment heuristic that adapts the window size to the current extend of concept change. The experimental results show, that accuracy, recall, and precision are well suited as indicators for concept changes

		$\gamma = 0.25$			$\gamma = 0.50$			$\gamma = 0.75$		
		$\alpha = 2.5$	$\alpha = 5.0$	$\alpha = 7.5$	$\alpha = 2.5$	$\alpha = 5.0$	$\alpha = 7.5$	$\alpha = 2.5$	$\alpha = 5.0$	$\alpha = 7.5$
PrTFIDF	Accuracy	86.76%	85.87%	81.53%	*87.11%*	86.82%	*80.93%*	86.87%	86.60%	82.34%
$\beta = 0.25$	Recall	94.69%	94.47%	87.81%	93.41%	93.60%	*85.80%*	94.74%	94.80%	88.88%
	Precision	65.02%	63.47%	57.43%	*66.13%*	65.56%	*56.87%*	65.44%	64.97%	58.62%
	Accuracy	87.01%	86.12%	81.52%	*87.11%*	**86.82%**	*80.93%*	86.89%	86.72%	82.34%
$\beta = 0.50$	Recall	*94.90%*	94.68%	87.81%	93.41%	**93.60%**	*85.80%*	94.70%	94.39%	88.88%
	Precision	65.41%	63.86%	57.43%	*66.13%*	**65.56%**	*56.87%*	65.46%	65.06%	58.62%
	Accuracy	86.99%	86.72%	82.47%	87.10%	87.01%	*80.93%*	86.98%	86.71%	82.47%
$\beta = 0.75$	Recall	94.33%	94.39%	88.62%	93.41%	93.36%	*85.80%*	94.33%	94.39%	88.62%
	Precision	65.53%	65.06%	58.89%	66.11%	65.93%	*56.87%*	65.51%	65.04%	58.89%
C4.5	Accuracy	89.95%	*90.05%*	*81.89%*	89.85%	90.00%	82.22%	89.63%	89.63%	83.10%
$\beta = 0.25$	Recall	70.81%	70.57%	*51.99%*	71.67%	71.42%	52.15%	71.09%	71.49%	54.10%
	Precision	*82.56%*	81.47%	*61.02%*	81.85%	82.07%	62.31%	81.41%	80.36%	64.13%
	Accuracy	89.87%	89.85%	83.08%	89.74%	**89.76%**	83.08%	89.63%	89.71%	83.08%
$\beta = 0.50$	Recall	*72.17%*	71.25%	53.58%	71.52%	**71.45%**	53.58%	71.09%	71.49%	53.58%
	Precision	81.31%	81.68%	65.33%	81.54%	**81.53%**	65.33%	81.41%	81.30%	65.33%
	Accuracy	89.82%	89.66%	83.08%	89.72%	89.66%	83.08%	89.60%	89.66%	83.08%
$\beta = 0.75$	Recall	71.97%	71.62%	53.58%	71.34%	71.62%	53.58%	70.96%	71.62%	53.58%
	Precision	81.10%	81.17%	65.33%	81.40%	81.17%	65.33%	81.24%	81.17%	65.33%

Table 6 Effect of varying the parameters α, β, and γ of the window adjustment heuristic on the performance of PrTFIDF and C4.5 in scenario B averaged over 4 trials (bold font indicates the configuration of the previous experiments, italic font indicates minimum and maximum values of the particular performance measure and learning method combination).

in text classification problems, and that recall and precision indicate concept changes more clearly than accuracy. Furthermore it could be observed that even using a very simple window of fixed size on the training data leads to significant performance improvements for all tested learning methods compared to learning on all previously seen examples. The proposed adaptive window management approach yields further performance improvements over the best approach with a window of fixed size for most of the learning methods. Hence both, the indicators for concept changes and the window adjustment heuristic based on them, provide promising starting points for future research and applications in adaptive information filtering.

References

1. James Allan. Incremental relevance feedback for information filtering. In *International ACM SIGIR Conference 1996*, Zürich, Swiss, August 1996.
2. Marko Balabanovic. An adaptive web page recommendation service. In *First International Conference on Autonomous Agents*, Marina del Rey, CA, February 1997.
3. Peter Clark and Robin Boswell. Rule induction with CN2: Some recent improvements. In *Machine Learning – Proceedings of the Fifth European Conference (EWSL '91)*. Springer, Berlin, Germany, 1991.
4. Thorsten Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of the 1997 International Conference on Machine Learning (ICML '97)*, 1997.
5. Ralf Klinkenberg. Maschinelle Lernverfahren zum adaptiven Informationsfiltern bei sich verändernden Konzepten. Master thesis, Universität Dortmund, Germany, February 1998.
6. Gerhard Kunisch. Anpassung und Evaluierung statistischer Lernverfahren zur Behandlung dynamischer Aspekte in Data Mining. Master thesis, Universität Ulm, Germany, June 1996.
7. W. Lam, S. Mukhopadhyay, J. Mostafa, and M. Palakal. Detection of shifts in user interests for personalized information filtering. In *International ACM SIGIR Conference 1996*, Zürich, Swiss, August 1996.
8. Carsten Lanquillon. Dynamic neural classification. Master thesis, Universität Braunschweig, Germany, October 1997.
9. Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-treshold algorithm. *Machine Learning*, 2:285–318, 1988.
10. Tom Mitchell. *Machine Learning*. McGraw Hill, New York, NY, USA, 1997.
11. J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, USA, 1993.
12. J. J. Rocchio Jr. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, USA, 1971.
13. Charles Taylor, Gholamreza Nakhaeizadeh, and Carsten Lanquillon. Structural change and classification. In *Workshop Notes on Dynamically Changing Domains: Theory Revision and Context Dependence Issues, 9th European Conference on Machine Learning (ECML '97), Prague, Czech Republic*, pages 67–78, April 1997.
14. Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, NY, USA, 1995.
15. Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23:69–101, 1996.