# Daimler Benz Research: System and Experiments

## Routing and Filtering

Thomas Bayer, Heike Mogg-Schneider, Ingrid Renz, Hartmut Schäfer
Daimler-Benz AG
Research and Technology
Wilhelm Runge Str. 11
D - 89081 Ulm
Germany
email: renz@dbag.ulm.daimlerbenz.com

## 1 General Approach

The retrieval approach is based on vector representation (bag of character strings), on dimension reduction (LSI - latent semantic indexing) and on statistical machine learning techniques in all processing levels. Two phases are distinguished, the *adaptation* phase based on training samples (texts) and the *application* phase, where each text is mapped to one or more categories (classes). The adaptation process is corpus dependent and automatic and, hence, domain and language independent.

The main idea of this approach is to generate different sets of simple features which represent different views to texts. For each text to be filtered/routed, different feature vectors are generated and classified into a decision vector which contains estimates of class membership probabilities. In the following step, these decision vectors are regarded as feature vectors and fed to another classifier that combines these set of decision vectors into the final one.

## 2 System Overview

Fig. 1 illustrates the principle design of the system (for more details about our work see [1] and [2]). First, resources are adapted using a text (training) corpus and deferred measurements of this corpus (steps (1) - (5)), such as feature and decision vectors; in these steps the algorithms described above are applied. After adaptation of all resources, the system is able to assign one or more categories along with probabilities to an unknown text.

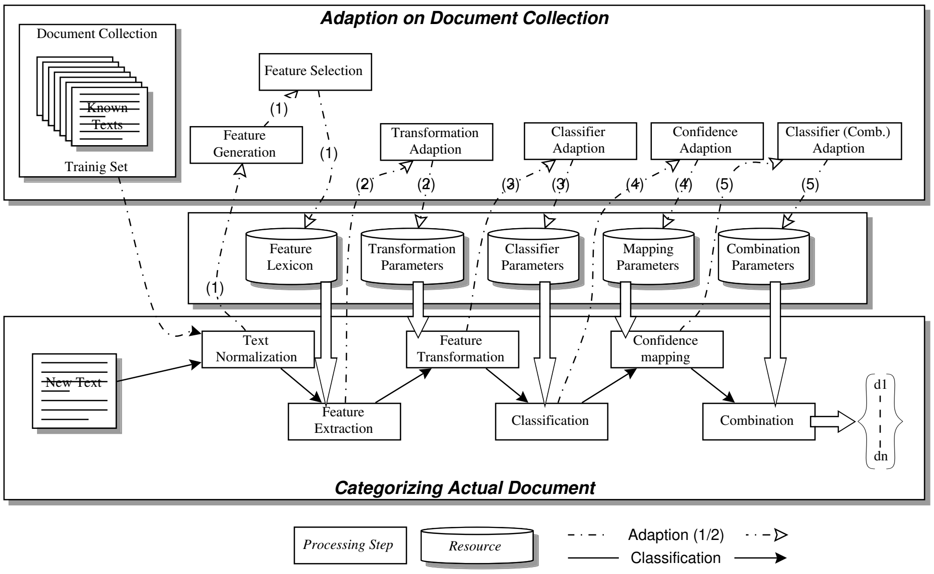



Figure 1: Adaptation and application are distinguished.

The final decision is represented as a decision vector. In case of routing the estimated probabilities $d_k$ of each category $k$ are sorted. In case of filtering – here, a binary decision has to be made – thresholds are calculated for each category; if the estimated probability is above the threshold, this category is assigned to the text, if it is below, the text is rejected with respect to this category.

We regard the TREC task as a 47-class problem, and therefore, construct classifiers with the ability to distinguish between 47 classes. An alternative approach would have been to construct 47 different 2-class classifiers each being responsible for one class and the complement of this class. However, this approach would have multiplied the computational effort in the optimization phase.

The systems for routing and filtering do not differ with respect to the processing steps except the final step of threshold comparison in case of filtering. However, the training samples and the number of samples for both tasks differed significantly. For both systems confidence mapping (see fig. 1 is not used because it did not significantly improve system's accuracy during evaluation.

# 3 Processing Steps

## 3.1 Textnormalization

The text is converted to a normalized form: all HTML tags are removed; all numbers converted to the digit 1; each word is converted to lower case; punctuation is removed. Each white-space character is replaced by the blank character.

## 3.2 Feature Generation

With two procedures (simple n-gram statistics and corpus-based sub-string computation), we computed different feature sets which represent different views to these texts. In every feature set, a feature is a string, i.e. a sequence of 3 or 4 characters (alphanumeric, blank), - this limitation of length is due to efficiency of text transformation. Every feature set only results from the normalized and complete texts of the training set. Neither topic descriptions nor external knowledge bases (thesauri, lexica) are used, but the information to which topic a text belongs.

**N-gram statistics**

For every topic, all word forms of the training texts are transformed into topic-specific lists of 3- and 4-grams together with their frequencies. Only frequent and topic-specific n-grams are selected as features.

For routing, we selected a set of 4242 3-grams and 5496 4-grams. For filtering, we chose 3820 3-grams and a set of 7596 4-grams.

**Corpus-based sub-string computation**

This approach acquires features by breaking words of the training set into parts (sub-strings) by an iterative method and collects them into a lexicon.

First, stop-words are defined as word forms which are very frequent and equally distributed among all topics. Then, affixes are computed as frequent beginnings and endings of word forms. Word forms which are not stop-words are split into smaller parts exploiting the morphological regularity that complex words are composed of simpler forms. If a complex form and one of its components are both members of the list of word forms, the complex string is divided into the component and the remaining character string.

The resulting sub-strings are ranked according to different selection criteria: chi-square, tf/idf-measure, correlation measure. For every topic, the first 50 sub-strings of these rankings are collected as features and transformed into 4-grams which are more tractable in the following step of text transformation.

The number of features in these sets are 831, 1995, and 4369 for routing and 2288, 3368, and 4146 for filtering.

Hence, we generated 5 feature sets for routing and 5 for filtering. They are different because the training texts have been different. For routing the dimension $L$ ranges from 831 to 5496, for filtering from 2288 to 7596.

## 3.3 Text Transformation

According to the five feature sets, every text is transformed into five vectors. For every feature, the text frequency is computed and inserted into an indexed vector $v_0$.

## 3.4 Dimension Reduction

The original dimension $L$ is reduced to a small number $L'$ of several hundreds. One well known method to reduce the vector space $R^L$ is the principal component analysis (PCA) which is based on the eigenvalues and eigenvectors of the covariance matrix of the sample vectors.

The reason for dimension reduction is that the higher $L$ is, the more training examples must be provided in order to avoid overfitting of the subsequent classifier to the training set.

In Information Retrieval, SVD (singular value decomposition) is typically used instead of the PCA for dimension reduction (*latent semantic indexing*, see [3]): $Y = UDV$, where $Y$ contains all feature vectors of the training set. It can be shown that the $L$ eigenvectors of the covariance matrix are identical to the vectors of matrix $U$. However, PCA requires less computation both in terms of time and space.

Depending on the original dimension of the vector space and the number of training samples we selected the most efficient way with respect to computational effort: if the number of samples $N$ is lower than the dimension of the feature space $L$ – which is mostly true for this input data – the Gram matrix of the feature vectors (dimension $N \times N$) should be used; otherwise, the covariance matrix is more appropriate (dimension $L \times L$).

### 3.5 Classification

The problem of routing/filtering is regarded as a classification task into $K$ categories ($K$-class problem). Hence, a classifier maps a feature vector into a $K$-dimensional decision vector where each component represents the a-posteriori probability that this feature vector belongs to category $k$. Another approach would have been to regard this problem as $K$ different problems in order to distinguish one class $k$ from all others. This approach would have led to $K$ 2-class problems.

For classification into one of $K$ categories (decision space), a numerical classification technique is employed. In [6] different numeric classifiers are compared, ranging from the standard Rocchio approach to linear and non-linear neural networks. The classification principle employed here is function approximation based on polynomials [5]. The $L'$ elements of $v \in R^{L'}$ (derived from $v_0$ by PCA) are combined by a polynomial function $x : v \rightarrow x(v)$ resulting in multiplicative combination of the elements. Mathematically, the polynomial classifier is defined as $d = A^T \times x(v)$, where $A \in R^{K \times X}$ is the coefficient matrix to be adapted and $X$ the dimension of the range of the function $x$. The coefficients are calculated by minimizing the mean-square error between the estimation $d$ and the true value $y$ – the target vector which is a unit vector with the 1 at the k-th position – describing the category membership of $v$:

$$E\{|A^T \times x(v) - y|^2\} = Minimum(A).$$

$E\{\ldots\}$ denotes the mathematical expectation In the optimization problem above $A$ is computed by regression assessing a training sample of size $N$ of pairs $(v^i, y^i)$. It can be shown ([5]) that the k-th element of $d$ estimates the a-posteriori probability $p(k|v)$. For a detailed description of the polynomial classifier design see [5] and [4].

The concept of the polynomial classifier has been extended with respect to $y$. $y$ is a unit vector indicating that an object belongs to exactly one class. For categorization tasks this assumption does not hold any more; therefore, $y$ contains as many 1's as class memberships exist. This includes some consequences for subsequent processing which are not discussed in detail here. However, the essential consequence is that the resulting decision vector can not be normalized to length 1.

The linear classifier is identical to the LLSF (linear least square fit) classifier described by Yang (see [7] and [8]). However, the mathematical principle is different in general if higher order polynomials are used. In this case, a non-linear function (e.g. quadratic polynomial) maps the feature space to the decision space yielding better separation of categories in the decision space.

## 4 Experiments

The relevant texts provided by NIST (24276 for routing, 4925 for filtering) have been divided into 75% training and 25% test texts for each of the 47 categories. With respect to fig. 1, the training texts represent the document collection from which all resources are derived. The full training text has been used for adaptation – typically everything between the HTML markers <TEXT> </TEXT> – although we noticed during evaluation that the relevant portion of a text may only be a fraction of the original text. For adaptation, only the training texts have been observed in all steps.

The test samples only served for optimization of parameter setting of each processing step. When accuracy scores are reported in the following, these scores have been obtained from runs on the test set.

All texts marked as relevant had been used for training and test. Non-relevant texts (marked as non-relevant and not marked and thus presumed to be non-relevant) have not been considered for adaptation, but some of them (2000 marked as non-relevant and 3000 not marked) have been included into the test set in case of filtering in order to determine the thresholds for rejects for each category. Topic files have not been used.

In optimizing the system's accuracy for routing/filtering, we experimented with different parameter settings for all processing steps except *Text Normalization*. In routing, the optimization criteria were (non-)interpolated average precision (IAP and NIAP in the following) for the (a-priori probability weighted) mean value of IAP and NIAP for all 47 categories. In filtering, we used the mean value of ASP, F1, and F2 for all 47 categories (equally weighted).

**Feature generation:** Several parameters could have been varied: the selection of n-grams (what is frequent - the most frequent 20%, 50% of all n-gram, what is topic-specific - in 5%, 20% of all topics), the selection criteria of the corpus-based sub-strings (information gain, mutual information, signal-to-noise-ratio) or the norming of feature vectors (binarization, normalization with max. value or to 1). Because of time constraints, none of these variations have been evaluated in detail.

**Dimension reduction:** For each of the 5 feature lexica the transformation matrix had been calculated based on the feature vectors of the training texts (the class membership knowledge is not used here). Depending on the original dimension of the vector space and the number of training samples we selected the most efficient way with respect to computational effort (see sect. 2.3), PCA of either the covariance or the Gram matrix.

In routing, the original dimensions of the feature space ranged from 831 to 5496. We experimented with reduced dimensions of 300-600-900-1200 for each of the 5 feature sets. We selected a dimension of 900 to reduce the different features to because the evaluation on the test texts showed that the accuracy raised up to 10% (in mean for all 5 feature sets) from 300 to 900, but did not raise in case of 1200. Additionally, a vector space of dimension 1200 may lead to an overfitting of the classifier because the size of the training set is below 20000 samples. The highest score for dimension 900 was 80% IAP and 84% NIAP for feature routing-1995.

Since the size of the training corpus was much smaller in case of filtering, we experimented in the range of 300-450-600-750 to reduce the original dimension which was in the range of 2288 and 7596. We decided to reduce to a dimension of 600 because the F1 and F2 values measured derived from the test set were maximized in this case. We obtained the best F1/F2 scores for feature filtering-3820 of 13 and 9 respectively.

Generating the coefficient matrix for dimension reduction is computational expensive (time and particular space); but calculation must be done only once for a given feature set. We used a Pentium 200MHz computer with 512MB RAM. This machine enables us to calculate eigenvectors from a matrix of size $6000 \times 6000$ (either Gram or covariance matrix). Transformation during application is fast, however.

**Classification:** We experimented with linear and quadratic polynomial classifiers and with a distance measuring (euclidean) classifier where each category was represented by the mean-value of its training vectors. Since the feature vectors are normalized to length 1, this reference classifier is identical to the Rocchio classifier using the cosine measure.

The evaluation showed that the polynomial classifier constantly outperformed the Rocchio approach. Finally, we decided to select linear polynomial classifiers for each of the five feature sets (for filtering and routing) because for the second order polynomial the number of parameters to adapt (the matrix $A$) grows quadratic with the dimension of the feature space and therefore, one would need more training data than the ones having been available. Hence, the experiments had shown that this classifiers are overfitted and have not been able to generalize on the test texts.

In order to overcome the problem of dimensionality, we also tried to reduce the dimension by PCA to a range of 50. However, the loss of information is too high in this case, which could not been compensated by the more powerful second order classification approach.

**Combination:** Each text has been represented by 5 different feature vectors resulting in 5 different decision vectors. All or a subset of them can be concatenated to a single vector which is regarded as a new feature vector of dimension $m \times 47$. A linear polynomial classifier is adapted on the training set of concatenated vectors resulting in the final decision vector for a text.

We experimented with different subsets in routing and filtering. The best scores have been achieved by combining all 5 decision vectors. Even the worst local subsystem improved the accuracy of an arbitrary subset of the remaining ones after combination with them. Without combination, the best local scores for IAP and NIAP have been 80.39% and 84.92% for the routing-1995 (LSI to 900) feature; combining all, the scores improved to 85.98% and 90.11% (note that the test texts only contained relevant texts). We detected the same behavior in case of filtering; we obtained the best local F1/F2 scores for feature filtering-3820 (LSI to 600) of 13 and 9 respectively. After combination they raised to 16 and 14 (in this case, 2/3 of the test text had been non-relevant ones).

**Threshold selection:** In case of filtering we selected thresholds for each category by maximizing the system's responses for the three different evaluation measures F1,F2 and ASP resulting in three 47-dimensional threshold vectors. Hence, this decision also has been adapted by the sample set. Since the F2 measure considers recall (penalizes missing relevant ones) the thresholds are lower than the F1 thresholds for all categories.

When applying the system, each decision vector of a text from the TREC-6 test set was compared against three different threshold vectors; the text was assigned to a category if the component of the decision vector exceeded the threshold component.

# 5 Results and Discussion

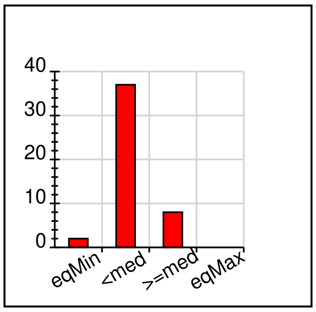
In the following some evaluations of our results for routing and filtering are presented as they were distributed from NIST.

## 5.1 Routing

The results for routing are summarized in the following two tables in fig. 2 which shows a comparison of our system with the other ones across the 47 topics; the left diagram represents the evaluation of the interpolated recall, the right one the precision at 1000 documents. The four bars in each diagram show how often the system fell to minimum, below median, above or equal median, or reached maximum.

The following tables show the results in detail:

```
Total number of documents over all queries
    Retrieved:    47000
    Relevant:      6872
```



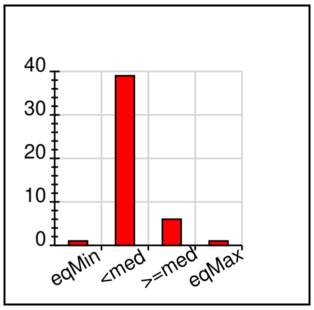



Figure 2: Comparison of our system with the other ones across the 47 topics; four values are displayed: the frequency the system achieved the minimum score, was below median, equal or above median and achieved maximum. The left one refers to interpolated recall, the right one to precision at 1000 documents.

```
    Rel_ret:       3334
Interpolated Recall - Precision Averages:
    at 0.00       0.6299
    at 0.10       0.4131
    at 0.20       0.3027
    at 0.30       0.2240
    at 0.40       0.1718
    at 0.50       0.1222
    at 0.60       0.0845
    at 0.70       0.0620
    at 0.80       0.0330
    at 0.90       0.0073
    at 1.00       0.0029
Average precision (non-interpolated) over all rel docs
                  0.1619
Precision:
  At    5 docs:   0.4213
  At   10 docs:   0.3830
  At   15 docs:   0.3645
  At   20 docs:   0.3553
  At   30 docs:   0.3326
```

```
  At  100 docs:   0.2436
  At  200 docs:   0.1820
  At  500 docs:   0.1102
  At 1000 docs:   0.0709
R-Precision (precision after R (= num_rel for a query)
docs retrieved):
    Exact:        0.2199
```

The results are disappointing. Three factors seem to be responsible for these unsatisfactory results:

1. The system had been trained with all relevant texts of the training set which comprises texts from quite different information sources, like AP, Wall Street Journal, etc. However, these texts are not representative for the test texts which come from the FBIS information source entirely. Another clue to this fact is that our filtering system which was trained with FBIS texts only and has a similar architecture was able to retrieve 3804 relevant texts compared to 3334 of the routing system. As a consequence the features trained do not necessarily match the ones of the FBIS texts.

2. We used all the text for training a text sample contained although we noticed during evaluation that only a small fraction of a text may really be relevant. Therefore, the feature space is further moved to a non-representative one.

3. The routing system was – mistakenly – evaluated by averaging the 47 topics due to their a-priori probability instead of averaging equally weighted (as NIST does in their evaluations). This affected the PCA and the classifier. Therefore, the optimum performance may not be reached.

## 5.2 Filtering

The filtering results are more encouraging as the figures for the evaluation measures show in fig. 3. The pooled results show the performance of the submitted results, the ranked bars show the potential improvement if optimum thresholds would be selected.

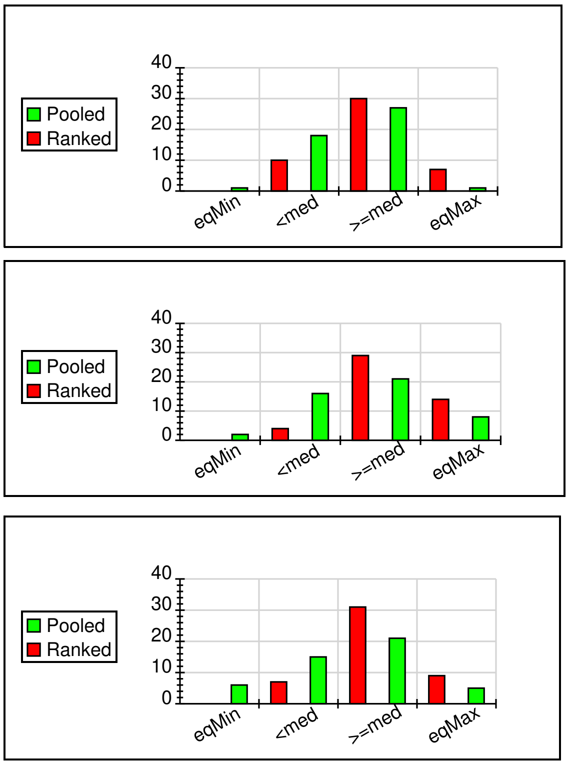



Figure 3: Comparison of our system with the other ones across the 47 topics; the same four values are displayed as in the previous figure. ASP, F1 and F2 are shown from top to bottom.

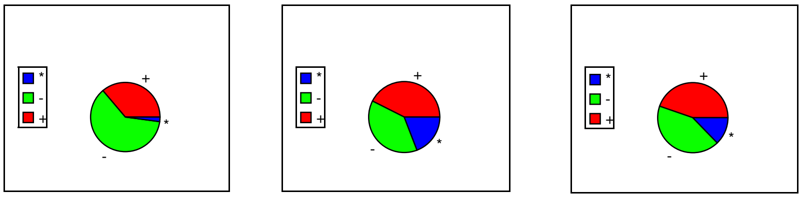


Figure 4: Evaluation of thresholds for ASP, F1 and F2. '+' means too many texts have been selected, '-' too few and '*' optimum.

The pooled scores of ASP, F1, and F2 are good: clearly, the scores lie above median. ASP and F1 are slightly better than F2; the reason is that the thresholds are smaller for F2 and hence, more documents are retrieved which are not relevant and therefore degrade the system's performance.

The ranked results in fig. 3 and the charts in fig. 4 both evaluate the threshold selection algorithm. The comparison between ranked results and pooled results shows that the thresholds are not optimum but reasonably good for F1 and F2. In these cases, roughly 20% of the thresholds are optimum and the remaining ones are equally covered by '+'s and '-'s. The threshold evaluation of ASP is dominated by '-'s. The reason for the non-optimum threshold is that the ASP histogram for a category with respect to different thresholds does not have a significant and clear peak which hints to the optimum threshold (e.g. histograms vary between 0.1 and 0.18). In contrast to ASP, the F1 and F2 histograms do have significant and clear peaks from which the optimum thresholds can clearly be extracted.

The scores in detail are listed here:

| | | F1 | | All Results | | |
|---|---|---|---|---|---|---|
| Topic | #rel | #docs | score | min | med | max |
| 1 | 51 | 2 | 6 | -318 | 9 | 56 |
| 3 | 76 | 89 | -33 | -676 | -6 | 28 |
| 4 | 80 | 109 | 27 | -696 | -12 | 27 |
| 5 | 7 | 1 | 3 | -781 | 0 | 4 |
| 6 | 165 | 153 | -171 | -913 | 35 | 112 |
| 11 | 174 | 67 | 71 | -156 | 52 | 82 |
| 12 | 292 | 234 | 102 | -15 | 102 | 381 |
| 23 | 7 | 0 | 0 | -621 | 0 | 0 |
| 24 | 42 | 1 | -2 | -69 | -7 | 1 |
| 44 | 4 | 0 | 0 | -1968 | -2 | 3 |
| 54 | 174 | 73 | 119 | -245 | 119 | 285 |
| 58 | 18 | 2 | -4 | -637 | -4 | 0 |
| 62 | 401 | 43 | 24 | -903 | 12 | 24 |
| 77 | 16 | 33 | -66 | -662 | 0 | 6 |
| 78 | 45 | 10 | 10 | -583 | 3 | 30 |
| 82 | 82 | 86 | -132 | -132 | 32 | 78 |
| 94 | 193 | 5 | 5 | -45 | 5 | 40 |
| 95 | 138 | 4 | -3 | -243 | -3 | 32 |
| 100 | 197 | 77 | 66 | -90 | 105 | 244 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 108 | 314 | 7 | 6 | -40 | 37 | 140 |
| 111 | 566 | 349 | 252 | -70 | 288 | 560 |
| 114 | 59 | 134 | -78 | -248 | 27 | 50 |
| 118 | 89 | 348 | -611 | -611 | -165 | 17 |
| 119 | 85 | 3 | -1 | -294 | -33 | -1 |
| 123 | 62 | 29 | -43 | -1533 | 0 | 36 |
| 125 | 27 | 0 | 0 | -1985 | 0 | 8 |
| 126 | 19 | 0 | 0 | -150 | 3 | 14 |
| 128 | 333 | 0 | 0 | -1780 | 0 | 49 |
| 142 | 229 | 963 | -1361 | -1361 | -257 | -10 |
| 148 | 260 | 25 | 70 | -881 | 0 | 70 |
| 154 | 175 | 1 | 3 | -1426 | 145 | 386 |
| 161 | 121 | 138 | -26 | -608 | 163 | 260 |
| 173 | 16 | 0 | 0 | -1995 | 0 | 1 |
| 180 | 17 | 0 | 0 | -1995 | -4 | 0 |
| 185 | 18 | 11 | -7 | -1352 | 0 | 10 |
| 187 | 21 | 161 | -307 | -317 | -165 | 0 |
| 189 | 890 | 548 | 209 | 0 | 180 | 837 |
| 192 | 7 | 0 | 0 | -556 | 0 | 0 |
| 194 | 4 | 0 | 0 | -1995 | -2 | 0 |
| 202 | 627 | 40 | 120 | -15 | 89 | 644 |
| 228 | 65 | 2 | 6 | -460 | 3 | 18 |
| 240 | 131 | 9 | -3 | -1980 | 1 | 12 |
| 282 | 28 | 0 | 0 | -68 | 0 | 9 |
| 10001 | 135 | 2 | 6 | -8 | 6 | 26 |
| 10002 | 321 | 1 | 3 | -383 | 6 | 40 |
| 10003 | 73 | 0 | 0 | -52 | 0 | 15 |
| 10004 | 18 | 47 | -89 | -89 | -2 | 9 |

| | | F2 | | All Results | | |
|---|---|---|---|---|---|---|
| Topic | #rel | #docs | score | min | med | max |
| 1 | 51 | 6 | -32 | -56 | -20 | 57 |
| 3 | 76 | 388 | -204 | -269 | -54 | 20 |
| 4 | 80 | 180 | -5 | -239 | -43 | 26 |
| 5 | 7 | 31 | -13 | -82 | -9 | 1 |
| 6 | 165 | 176 | -211 | -211 | -47 | 120 |
| 11 | 174 | 317 | 59 | -171 | -5 | 62 |
| 12 | 292 | 468 | 135 | -293 | 65 | 422 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 23 | 7 | 0 | -7 | -16 | -8 | -7 |
| 24 | 42 | 1 | -43 | -69 | -48 | -25 |
| 44 | 4 | 0 | -4 | -50 | -5 | 5 |
| 54 | 174 | 76 | 20 | -219 | 76 | 311 |
| 58 | 18 | 2 | -20 | -27 | -20 | -3 |
| 62 | 401 | 76 | -282 | -435 | -349 | -203 |
| 77 | 16 | 53 | -69 | -69 | -16 | -6 |
| 78 | 45 | 11 | -21 | -57 | -21 | 21 |
| 82 | 82 | 89 | -131 | -137 | -2 | 89 |
| 94 | 193 | 60 | -118 | -224 | -185 | -87 |
| 95 | 138 | 195 | -238 | -338 | -139 | -14 |
| 100 | 197 | 222 | -34 | -217 | 26 | 196 |
| 108 | 314 | 59 | -198 | -334 | -184 | 240 |
| 111 | 566 | 1000 | 89 | -576 | 84 | 621 |
| 114 | 59 | 117 | 9 | -128 | 8 | 50 |
| 118 | 89 | 732 | -701 | -701 | -209 | -47 |
| 119 | 85 | 17 | -92 | -215 | -102 | -72 |
| 123 | 62 | 42 | -79 | -79 | -49 | 24 |
| 125 | 27 | 0 | -27 | -56 | -27 | -8 |
| 126 | 19 | 0 | -19 | -37 | -4 | 16 |
| 128 | 333 | 0 | -333 | -1113 | -318 | -220 |
| 142 | 229 | 1000 | -664 | -664 | -327 | -126 |
| 148 | 260 | 226 | 344 | -286 | -256 | 344 |
| 154 | 175 | 0 | -175 | -180 | 167 | 405 |
| 161 | 121 | 166 | -7 | -174 | 164 | 249 |
| 173 | 16 | 17 | -23 | -42 | -20 | -9 |
| 180 | 17 | 0 | -17 | -72 | -19 | -17 |
| 185 | 18 | 10 | -18 | -330 | -18 | -4 |
| 187 | 21 | 681 | -642 | -642 | -176 | -21 |
| 189 | 890 | 1000 | -35 | -886 | -146 | 923 |
| 192 | 7 | 0 | -7 | -40 | -14 | -7 |
| 194 | 4 | 0 | -4 | -84 | -7 | -4 |
| 202 | 627 | 566 | 202 | -627 | -374 | 834 |
| 228 | 65 | 38 | -58 | -220 | -43 | -2 |
| 240 | 131 | 31 | -112 | -449 | -118 | -87 |
| 282 | 28 | 0 | -28 | -50 | -27 | -16 |
| 10001 | 135 | 0 | -135 | -158 | -120 | -76 |
| 10002 | 321 | 26 | -247 | -326 | -286 | -192 |
| 10003 | 73 | 0 | -73 | -648 | -73 | -18 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 10004 | 18 | 39 | -52 | -66 | -19 | -2 |

| | | ASP | | All Results | | |
|---|---|---|---|---|---|---|
| Topic | #rel | #docs | score | min | med | max |
| 1 | 51 | 3 | 0.059 | 0.000 | 0.098 | 0.386 |
| 3 | 76 | 308 | 0.103 | 0.000 | 0.118 | 0.260 |
| 4 | 80 | 133 | 0.235 | 0.000 | 0.086 | 0.260 |
| 5 | 7 | 1 | 0.143 | 0.000 | 0.082 | 0.286 |
| 6 | 165 | 153 | 0.029 | 0.000 | 0.150 | 0.236 |
| 11 | 174 | 224 | 0.217 | 0.000 | 0.135 | 0.227 |
| 12 | 292 | 744 | 0.215 | 0.000 | 0.202 | 0.416 |
| 23 | 7 | 0 | 0.000 | 0.000 | 0.000 | 0.080 |
| 24 | 42 | 1 | 0.000 | 0.000 | 0.006 | 0.072 |
| 44 | 4 | 6 | 0.042 | 0.000 | 0.004 | 0.375 |
| 54 | 174 | 77 | 0.226 | 0.000 | 0.031 | 0.524 |
| 58 | 18 | 2 | 0.000 | 0.000 | 0.000 | 0.045 |
| 62 | 401 | 41 | 0.029 | 0.000 | 0.010 | 0.045 |
| 77 | 16 | 1000 | 0.001 | 0.000 | 0.000 | 0.125 |
| 78 | 45 | 1000 | 0.045 | 0.000 | 0.050 | 0.235 |
| 82 | 82 | 90 | 0.011 | 0.000 | 0.148 | 0.325 |
| 94 | 193 | 1000 | 0.032 | 0.000 | 0.021 | 0.072 |
| 95 | 138 | 5 | 0.001 | 0.000 | 0.017 | 0.153 |
| 100 | 197 | 245 | 0.133 | 0.000 | 0.197 | 0.398 |
| 108 | 314 | 408 | 0.157 | 0.000 | 0.148 | 0.246 |
| 111 | 566 | 759 | 0.198 | 0.000 | 0.233 | 0.394 |
| 114 | 59 | 106 | 0.219 | 0.000 | 0.187 | 0.305 |
| 118 | 89 | 543 | 0.007 | 0.000 | 0.056 | 0.097 |
| 119 | 85 | 3 | 0.004 | 0.000 | 0.031 | 0.053 |
| 123 | 62 | 44 | 0.009 | 0.000 | 0.022 | 0.215 |
| 125 | 27 | 0 | 0.000 | 0.000 | 0.000 | 0.083 |
| 126 | 19 | 15 | 0.004 | 0.000 | 0.022 | 0.237 |
| 128 | 333 | 9 | 0.003 | 0.000 | 0.006 | 0.056 |
| 142 | 229 | 1000 | 0.056 | 0.000 | 0.128 | 0.173 |
| 148 | 260 | 212 | 0.482 | 0.000 | 0.008 | 0.511 |
| 154 | 175 | 1 | 0.006 | 0.000 | 0.274 | 0.706 |
| 161 | 121 | 168 | 0.149 | 0.000 | 0.455 | 0.692 |
| 173 | 16 | 37 | 0.007 | 0.000 | 0.000 | 0.087 |
| 180 | 17 | 26 | 0.000 | 0.000 | 0.000 | 0.001 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 185 | 18 | 15 | 0.059 | 0.000 | 0.000 | 0.148 |
| 187 | 21 | 147 | 0.003 | 0.000 | 0.006 | 0.060 |
| 189 | 890 | 916 | 0.153 | 0.000 | 0.129 | 0.329 |
| 192 | 7 | 0 | 0.000 | 0.000 | 0.000 | 0.010 |
| 194 | 4 | 1 | 0.250 | 0.000 | 0.000 | 0.250 |
| 202 | 627 | 673 | 0.216 | 0.000 | 0.137 | 0.415 |
| 228 | 65 | 11 | 0.013 | 0.000 | 0.045 | 0.117 |
| 240 | 131 | 11 | 0.006 | 0.000 | 0.015 | 0.051 |
| 282 | 28 | 0 | 0.000 | 0.000 | 0.000 | 0.107 |
| 10001 | 135 | 2 | 0.015 | 0.000 | 0.015 | 0.074 |
| 10002 | 321 | 3 | 0.009 | 0.000 | 0.008 | 0.048 |
| 10003 | 73 | 0 | 0.000 | 0.000 | 0.000 | 0.068 |
| 10004 | 18 | 23 | 0.000 | 0.000 | 0.000 | 0.167 |

**Acknowledgement** Ulrich Kressel and Jürgen Franke, both members of our research group, helped a lot in computing the PCA for large matrices and with fruitful discussions.

## References

[1] T. Bayer, U. Bohnacker, I. Renz, Information Extraction From Paper Documents, *Handbook of Optical Character Recognition and Document Image Analysis* (P.S.P. Wang and H. Bunke, Eds.), Singapore, 1997.

[2] T. Bayer, U. Kressel, H. Mogg-Schneider, and I. Renz, Categorizing Paper Documents - A Generic System for Domain and Language Independent Text Categorization, to appear in *Journal of Computer Vision and Image Understanding. Special Issue on Document Image Understanding and Retrieval*, 1998.

[3] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman, Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science*, 41(6), 1990.

[4] U. Kressel and J. Schürmann, Pattern Classification Techniques Based on Function Approximation, *Handbook of Optical Character Recognition and Document Image Analysis* (P.S.P. Wang and H. Bunke, Eds.), Singapore, 1997.

[5] J. Schürmann, *Pattern Classification*, Wiley, New York, 1996.

[6] H. Schütze, D. Hull, and J. Pederson, A Comparison of Classifiers and Document Representation for the Routing problem, *Proceedings, 18th Int. ACM-SIGIR Conf. on Research and Development in Information Retrieval*, Seattle, WA, 1995.

[7] Y. Y. Yang and C. G. Chute, An Application of Least Squares Fit Mapping To Text Information Retrieval, *Proceedings, 16th Int. ACM-SIGIR Conf. on Research and Development in Information Retrieval*, Pittsburgh, PA, 1993.

[8] Y. Y. Yang, Noise reduction in a statistical approach to text categorization, *Proceedings, 18th Int. ACM-SIGIR Conf. on Research and Development in Information Retrieval*, Seattle, WA, 1995.