Microsoft Cambridge at TREC 2002: Filtering track

S E Robertson* S Walker H Zaragoza R Herbrich

Microsoft Research Ltd, 7 J.J.Thomson Avenue, Cambridge CB3 0FB, UK
email {ser,sw,hugoz,rherb}@microsoft.com

1 Summary

Six runs were submitted for the Adaptive Filtering track, four on the adaptive filtering task (ok11af??), and two on the routing task (msPUM?). The adaptive filtering system has been somewhat modified from the one used for TREC–10, largely for efficiency and flexibility reasons; the basic filtering algorithms remain similar to those used in recent TRECs. For the routing task, a completely new system based on perceptrons with uneven margins was used.

2 Okapi at TRECs 1–10

A summary of the contributions to TRECs 1–7 by the Okapi team, first at City University London and then at Microsoft, is presented in [6]. In TRECs 7–10 we took part in the adaptive filtering track, initially concentrating on the thresholding problem, but by TREC–9 we had a full adaptive filtering system with query expansion as well as adaptive thresholding. This adaptation could be used to optimise performance on a number of effectiveness measures and produced good results on both the TREC–9 measures, linear utility and the 'precision-oriented' measure, but performed poorly on the Reuters topics at TREC–10. In earlier TRECs on various adhoc tasks we had concentrated on the weighting schemes and pseudo relevance feedback (blind feedback), and had developed the successful BM25 weighting function but had had only limited success with blind feedback.

3 Adaptive Filtering

3.1 Okapi systems

At the Microsoft Research laboratory in Cambridge, we are developing an evaluation environment for a wide range of information retrieval experiments. This environment is called Keenbow. The various Okapi systems discussed below are seen as components of Keenbow. Many aspects of the systems, including the weighting scheme and the query expansion methods used, reflect the various components of the probabilistic model of retrieval discussed at length in [9].

The Okapi Basic Search System (BSS), which has been used in all Okapi and Okapi/Keenbow TREC experiments up to TREC–9, is a set-oriented ranked output system designed primarily for probabilistic-type retrieval of textual material using inverted indexes. There is a family of built-in weighting functions collectively known as BM25, as described in [5, Section 3] and subsequent TREC papers. In addition to weighting and ranking facilities it has the usual boolean and quasi-boolean (positional) operations and a number of non-standard set operations. Indexes are of a fairly conventional inverted type. Preprocessing includes stopping and stemming and matching a small exceptions dictionary (selected phrases, synonyms and words marked as not suitable for query expansion).

The primary method of using the BSS in adaptive filtering upto TREC–10 was to accumulate small batches of documents, index each batch as a separate BSS database, and search the profiles against it. This was not a very efficient process, and has some limitations – for example, adaptation could only be between batches (according to the TREC filtering track rules). For TREC 2002, we developed a new Okapi/Keenbow component called the Basic Filtering Dogsbody (BFD). The primary principle of the BFD is that a database of profiles (queries) is maintained, and each incoming document is searched against this database. In some sense this makes it a true filtering system, as opposed to an adhoc search system adapted for filtering. The BFD itself does not maintain a cumulative database of documents, but does keep up-to-date the dictionary part of such a cumulative database, consisting of terms and collection frequencies.

Adaptive methods are divided into query expansion or modification and threshold adaptation. Query expansion is performed by the BFD, on the basis of the text query and the cumulated set of known relevant documents (the most recent ones only if there are many). Threshold adaptation is performed by a script built on top of the BFD. This normally involves a search on the reference database, i.e. the cumulative database of all documents received so far. This is a conventional BSS database, and as previously is built in batches (and is therefore not completely up-to- date). Other aspects of the filtering operation, including the history and current state of the profile, are also built as scripts. The master script defines a set of rules for triggering the adaptive procedures; for TREC 2002, the main trigger for updating a profile (query expansion and threshold adaptation) is the retrieval of a relevant

*Also at City University, London, UK.

document. In the experiments described, this happens at every relevant document, and is immediate (i.e. before the next incoming document is processed). The same procedures are triggered occasionally for documents that have failed to retrieve a relevant document for some time.

The adaptive filtering runs were done on a 550MHz Xeon (512KB Cache) with 2Gb RAM and a Dell with two 400 MHz Pentium processors and 512 Mb. Both machines were running Solaris 7. The network was 100Mbps ethernet.

3.2 Algorithms and parameters

Reports from the last two years [7, 8] contain fairly detailed accounts of the filtering system and the adaptation methods used, in particular the relation between the optimisation measures and the threshold. In respect of the algorithms used, this year's system is very similar to last year's; Table 1 is an attempt to summarise the large number of parameters used. Essentially these parameters were set by a series of tuning experiments on the OHSU filtering database (the OHSUMED test collection, adapted for the filtering task for TREC–9, with the OHSU topic set). While this collection is rather different from the Reuters collection, the intention was to look for parameters that would be generally good, rather than ones that would be highly tuned to a particular database. This aim will be furthered by later work on this year's collection, to see how far from optimal the chosen values are. The one parameter which was adjusted from its best value for OHSU was the target number of documents for initial threshold setting. Since this parameter is an absolute number to be retrieved over the entire test set, it is highly dependent on test set size – in fact it would be better expressed as a proportion or probability than as an absolute number. However, on top of this consideration, the OHSU tuning suggested a rather lower value for utility optimisation than for fbeta optimisation.

3.3 Overview of the filtering procedure

At a particular iteration of the process, any query modification needs to take place before any threshold setting. It may also be necessary, after query reformulation but before threshold setting, to recalculate the scores of the previously-retrieved documents, for the adaptation of β.

The document collection is processed a document at a time. If a document is retrieved for any profile, it is immediately checked for relevance. If relevant, the query is updated and then the threshold is updated. At intervals defined by the batch size indicated in the table, the reference database is updated with all documents which have arrived since the last batch. Also, any profile that has not been updated since the last batch is updated.

3.4 Filtering results

As with the official track results, the measures reported are T11SU (scaled utility), T11F (Fbeta measure with beta=0.5), set precision and set recall.

Four runs were submitted, labelled ok11af[ls][ub]. Those with final letter u were optimised for T11SU, and those with final letter f for T11F. The next-to-last letter represents the source of the text topics – l (long) indicates the full text (title, description and narrative), and s (short) denotes title only. In common with other participants, we found very large differences between our performance on the assessor and intersection topics.

The results shown in Table 2 relate to assessor topics only. They are also very slightly different from the official runs, following discovery of a small bug in the system used. Evaluation is based on the full relevance judgements used for the official evaluation. For the runs corresponding to the official runs, adaptation is based on the relevance judgements available for that purpose. Additional runs were made using all relevance data for adaptation. The coding of the runs is:

lms long, medium or short initial topics (medium = title + description)

ub optimised for utility or FBeta

OR adaptation using original or complete relevance judgements

Disappointingly, the runs optimised for utility do marginally better on the FBeta measure than the run optimised for FBeta, at least when using the original relevance data. (This is the exact opposite of the result for last year!). It seems that the method for setting thresholds for FBeta, which involves estimating the total relevant in the collection, is producing somewhat erratic results. Further diagnostic testing is required.

Starting with longer topics may help a little (on utility at least) but the differences do not seem consistent (medium length topics seem to have no advantage over short ones). It seems from the assessor topic results at least that it is possible for an adaptive filtering system to bootstrap its performance reasonably well even if the starting point is not very good.

Intersection topics

However, it is difficult to reconcile the tentative conclusion above with the terrible performance on the intersection topics. One possible suggestion is that the 'relevance' judgements for the intersection topics (i.e. the assignment of documents to two different topic codes by Reuters editorial staff) fail to define a set of related documents with the sort of coherence that we find in assessor relevance judgements. Another is that the pairs of topics may have been unbalanced in some way, leaving it difficult for the filtering systems to infer criteria covering both aspects.

For the run corresponding to ok11aflu, the results are: T11SU=0.251, T11F=0.040, Precision=6.6%, Recall=2.3%. All the others are similarly bad or worse. We looked in detail at two topics, R195 and R181. R195 is formed by the intersection of Reuters topic categories GVOTE (Elections)

Table 1: Parameters for adaptive filtering

See notes below and [7, 8] for explanations of these parameters	
BM25 parameters:	
k_1	1.3
b	0.55
Score calibration:	
These parameters define the mapping from Okapi score to probability of relevance – $P(R)$ is estimated as a linear function of score, slope Gamma and intercept Beta. At each threshold updating, Beta (but not Gamma) is re-calibrated using scores of documents of known relevance. The 'mythical reldocs' serve as a Bayesian prior in this re-calibration.	
Initial beta	-0.66
Mythical reldocs for beta re-calibration	3
Gamma	2.9
Threshold adaptation:	
Initially, the threshold is set at a level estimated to retrieve a certain target number of documents over the whole test set. As relevant documents are retrieved, the threshold is moved up a ladder until it reaches the level defined by optimising the required parameter.	
Initial target no. of documents (FBeta)	70
Initial target no. of documents (Utility)	25
Ladder step	2
Query modification:	
Query modification uses the last n relevant documents retrieved (including the training sample if necessary), together with the original text query. Terms are ranked by absolute term selection value (new offer weight). All those exceeding the threshold are chosen, subject to both a minimum and a maximum number of terms.	
Reldocs used for modification	20
Maximum terms	25
Minimum terms	3
Absolute term selection value threshold	2
Document batching:	
Determines how often the accumulated reference database is updated, and also how often the threshold updating procedure is initiated for profiles which have retrieved no relevant documents since the last such update.	
Batch size	50,000

Further notes on thresholding

For Utility, the threshold calibrated as a log-odds probability is raised by one ladder-step for each relevant document retrieved. This is then compared with the level defined by the utility function, and the lower of the two is chosen. After 8 relevant documents have been retrieved, the level defined by the utility function is always chosen.

For Fbeta, a similar procedure is followed, but instead of the level defined by the utility function, the estimated optimum Fbeta threshold is used.

The ladder function is different from last year. The target is reduced pro-rata according to the estimated remaining number of documents to come, and then further divided by $((ladder\ step)**(numberofrelevantdocuments))$.

Thus if the ladder step is set to 1, the ladder is effectively switched off. Higher values give larger steps.

Table 2: Main results

Topics	Relevance judgements used for adaptation	Corresponding official run	T11SU	T11F	Precision	Recall
Utility optimisation						
long	original	ok11aflu	0.435	0.421	49.9	34.4
long	all		0.439	0.419	46.8	37.8
medium	original		0.405	0.405	48.5	31.9
medium	all		0.412	0.405	46.8	34.4
short	original	ok11afsu	0.406	0.404	48.2	33.0
short	all		0.418	0.413	46.2	36.7
FBeta optimisation						
long	original	ok11aflb	0.405	0.394	52.4	26.1
long	all		0.410	0.405	50.4	28.4
medium	original		0.396	0.392	52.0	26.3
medium	all		0.411	0.415	50.6	29.7
short	original	ok11afsb	0.404	0.393	52.0	25.9
short	all		0.418	0.411	50.8	29.0

Table 3: Titles of relevant and retrieved documents, topic R195

Training relevant:	1	Churches put poverty on NZ election agenda
	2	Dole accuses Clinton of “mediscare” ad campaign
	3	Clinton blocks federal loans to deadbeat parents
Test relevant:	4	Florida’s elderly key to Dole campaign
	5	U.S. group seeks child food-aid support
	6	Poverty is toughest task for next Nicaraguan leader
	7	Dole visits Florida, promises to save medicare
	8	Relaxed, confident Clinton stumps in central Florida
	9	Clinton would mull law aiding retirees if elected
	10	Arizona voters back lottery measure
	11	NZ’s National, Labour agree to pension referendum
	12	Poland’s pension reform under election cloud
	13	UK’s Dorrell details old age care insurance plan
	14	UK welfare reform to head Major’s election agenda
	15	Polish Solidarity sees growth as top economic goal
Retrieved:	16	S. Africa releases conservative welfare blueprint
	17	NYC agency says welfare poses big budget challenge
	18	The inexorable GST [Australian sales tax]
	19	New Moldovan leader seen backing market reforms
	20	Despite good times, many in U.S. need charity
	21	HUD chief warns U.S. near housing crisis for poor
	22	Study finds up to 10% of Swiss are poor
	23	UK’s Blair to unveil welfare plans
	24	British magazine offers help to homeless
	25	French government approves anti-poverty plan
	26	UK Labour’s Brown vows no tax and spend cure-all
	27	French MPs debate controversial anti-poverty bill

Table 4: Titles of training relevant documents, topic R181

Training relevant:	1	FoxMeyer Drug declares bankruptcy after sale falls through
	2	Foxmeyer says drug unit files for bankruptcy
	3	Westa receiver seeks Prochnik manager

and GWELF (Welfare, Social Services); R181 from C16 (Insolvency/liquidity) and C411 (Management Moves). In both these cases, as in many other intersection topics, there is no overlap at all between test relevant and retrieved: recall, precision, FBeta, unnormalised utility are all zero.

For topic R195, titles of the 3 training documents for adaptive filtering are given in Table 3, together with most of the relevant documents from the test set, and most of those retrieved in run ok11aflu (a few, including some duplicates, have been left out in the interests of saving space).

It may be seen that the documents found by the system are broadly in the right area – some look less obviously good candidates than others, but there are several in the list which one might reasonably expect to be relevant. One issue is that it seems that in order to qualify for the Election & Welfare category in Reuters, a document has to relate to a particular election. This probably excludes some of the retrieved documents, but not for example number 23, which does indeed relate to the impending British general election, exactly as do 13 and 14. However, 23 was assigned (in addition to GWELF) the code GPOL (Domestic Politics) but not GVOTE. One can only conclude that in this instance at least, the Reuters coding is just not very consistent. Number 26 is even worse – it has various headings relating to economics and finance, and GPOL and GCAT (Government/Social), but not GVOTE (despite the fact that it reports a campaign speech by someone not then in government) and not GWELF (despite the fact that a significant part is about poverty and unemployment).

We might have hoped to retrieve at least some of the relevant set. However, the filtering system is quite sensitive to adaptation – if it is getting no encouragement (in the form of positive relevance judgements) it will keep the threshold very high (the penalties for allowing through much more are too great).

In the case of topic R181, we show just the three training examples in Table 4

In this case, two of the titles relate to the same story. The interpretation of Reuters topic C411 (Management Moves) is supposed to be moves such as management appointments or resignations. Number 1 has a brief mention of an appointment in a story about the bankruptcy of FoxMeyer; number 2 is essentially an abbreviated version of no. 1, though the appointment part has been retained. Number 3 has (in our interpretation) no management moves in the sense given at all: the receiver is seeking not an individual but a financial institution to manage and sell a stake in another company. The ok11aflu run retrieved 12 documents, all squarely in the insolvency area, but none containing management moves. (Several of them relate to FoxMeyer, but there is also a group relating to Bulgarian banks. The one Bulgarian bank story which was marked as relevant was not selected in ok11aflu.) Thus this example seems to be an instance of one of the two original Reuters categories dominating. However, part of the reason is the choice of positive examples for training – it is certainly the case that those particular examples emphasise only one of the Reuters categories.

Reuters categories are often very broad concepts, and must be hard to assign consistently. On the evidence of these two cases, one might suggest that the intersection operation, together with the accidental choice of training examples, has significantly compounded the noise.

4 Routing

The perceptron-based system was developed for the TREC routing task independently of Okapi. The theoretical work leading to this model was carried out in 2001 and first evaluations on smaller datasets (such as Reuters-21578) were carried out at the beginning of this year [4]. Our TREC 2002 runs constituted the first full-scale implementation and evaluation of this model.

Research on Perceptrons is motivated by the recent success of soft-margin support vector machines for routing [3]. Soft-margin support vector machines are high-dimensional linear classifiers that maximise a quantity called the *margin* while keeping the training error close to zero. Because of the intimate relationship between margin and generalisation error, maximising the former will (asymptotically) minimise the latter.

When the training set is not linearly separable in its feature space the margin is maximised while allowing a small number of misclassification errors. The *cost* of a misclassification is determined prior to training by a learning parameter, C. An additional parameter, j, is used to weight differently positive and negative misclassifications. These two parameters are set in general by k-fold cross-validation ([3]).

Different theoretical and practical reasons made us search for alternative solutions to the SVM for the task of document routing:

1. It is theoretically not clear under which conditions large margin classifiers may lead to good *rankings* (as opposed to good classification).

2. There are other linear classifiers which do not maximise the margin but perform as well as the SVM for many classification tasks. Generalisation error bounds for these algorithms exist and some are tighter than those of the soft margin SVM.

3. Training times for SVMs are extremely long.

4. The need to optimise C and j multiplies the number of times we need to train the systems.

In particular, the perceptron learning algorithm (PLA) is a fast learning algorithm for linear classifiers, and it has been shown recently that it shares with the SVM some strong theoretical properties. In particular, one can show that *sparsity* for the perceptron (roughly speaking, the number of training updates) works similarly to margin for the SVM, that is, high sparsity guarantees low generalisation error. Furthermore, it has been shown that the existence of a large margin solution

implies the high sparsity of perceptron solutions. This means, again roughly speaking, that if there exists a good SVM solution (that is, one with a large margin) then the perceptron solution on the same dataset is likely to be good as well (see [1] [4] for a more formal discussion of these topics).

Our initial experiments in routing with the PLA (using the Reuters-21578 topics collection, and the average precision performance measure) showed that although it was slightly outperforming the SVM for topics with many positive examples, it underperformed significantly for smaller topics. This seems to indicate that one needs to impose some margin constraints on very small topics.

The margin-PLA [2] is a modified PLA which guarantees a solution with a minimum margin, i.e. the resulting margin is within a factor of $\tau/(2\tau+1)$ of the maximum possible margin (which would be found by an SVM). τ is therefore a parameter (similar to C) which must be set prior to training. While experiments with the margin-PLA showed improvement in performance over the PLA for small topics, it greatly increased the training time and decreased the sparsity of the solution. One of the reasons for this is that the margin constraints are symmetrical, that is, if we wish to enforce a large margin with respect to the relevant documents, we must do the same with respect to the irrelevant documents — a task that is too expensive because of their large number.

For these reasons, we modified the margin-PLA algorithm to take account of the asymmetry of the problem, and we replaced the constant τ by two constants, τ_{+1} and τ_{-1}, which enforce different margins with respect to the relevant $(+1)$ and irrelevant (-1) documents. This led to a great improvement of the speed of the training algorithm and the sparsity of the resulting solutions. Furthermore, when we optimised by cross validation the parameters τ_{+1} and τ_{-1} the resulting solutions outperformed the SVM on Reuters-21578 [4].

In the following sections we describe our algorithm, the perceptron learning algorithm with uneven margins (or PLAUM), its implementation for the TREC 2002 routing task, and summarise the results obtained.

4.1 The PLAUM algorithm

We present in Algorithm 1 the PLAUM as implemented for our TREC 2002 routing experiments. Basically, we iterate over the training sample testing for every pattern $\vec{x}_i$ if the output of our classifier $(\langle \vec{w}, \vec{x}_i \rangle + b)$ is of the right sign and, even more, greater than the required factor on the minimal margin for the pattern's class y_i, τ_{y_i}. When *all* the patterns satisfy this condition, the algorithm stops.

Despite the high dimension of documents (from hundreds to tens of thousands) linear separability cannot always be guaranteed. This condition can be relaxed by the so-called λ-*trick*, which extends each document vector $\vec{x}_i$ by a vector of size m with value λ for the ith coordinate and zero elsewhere (m is the number of training documents). To implement this it suffices to redefine the inner-product function as: $\langle \vec{w}, \vec{x}_i \rangle := \sum_{j=1}^{m} w_j x_{i,j} + w_{m+i}\lambda$.

The PLAUM algorithm with the λ-trick is guaranteed to always stop at a solution if $\lambda > 0$. Nevertheless, in some pathological cases the algorithm can iterate a very large number of times. For this reason we include the parameter T which sets a maximum to the number of epochs (iterations over the training set) allowed.

Finally, for completeness we have included in the algorithm the learning parameter η. However, in our experiments this parameter was always set to 1.

Algorithm 1 PAUM $(\tau_{-1}, \tau_{+1}, T, \eta, \boldsymbol{z})$

Require: A linearly separable training sample
  $\boldsymbol{z} := (\boldsymbol{x}, \boldsymbol{y}) \in (\mathcal{X} \times \{-1, +1\})^m$
Require: A learning rate $\eta \in \mathbb{R}^+$
Require: A maximum epochs parameter T
Require: Two margin parameters $\tau_{-1},\ \tau_{+1} \in \mathbb{R}^+$
  epoch $\leftarrow 0$; $i \leftarrow 1$; updated $\leftarrow m$
  $\vec{w} \leftarrow \vec{0}$; $b = 0$; $R \leftarrow \max_{\vec{x}_i \in \boldsymbol{x}} \|\vec{x}_i\|$
  repeat
    if $y_i\ (\langle \vec{w}, \vec{x}_i \rangle + b) \leq \tau_{y_i}$ **then**
      $\vec{w} \leftarrow \vec{w} + \eta y_i \vec{x}_i$
      $b \leftarrow b + \eta y_i R^2$
      updated $\leftarrow i$
    end if
    $i \leftarrow i + 1$
    if $(i > m)$ **then**
      $i \leftarrow 1$; epoch $\leftarrow$ epoch+1
    end if
  until $(i =$ updated$)$ **or** (epoch $\geq T$)
  return $(\mathbf{w}, b)$

4.2 Data representation

We considered two different representations of the documents: the usual *tf*×*idf* representation and a BM25-based representation where *idf*'s are replaced by topic-dependent BM25 weights.

Pre-processing was kept to a minimum: no stemming was used nor were stop words removed. Punctuation marks and letter case were removed, and all character strings appearing in fewer than three documents were eliminated. All other character strings became features (terms) of the linear classifier.

For the *tf*×*idf* representation all resulting features in the training set were considered (approximately 10 000). For the BM25-based representation only features in relevant documents were considered (approx. 600 on average). Finally all vectors were normalised to have unit Euclidean norm.

4.3 Model Selection

Two parameters need to be set prior to training: τ_{+1} and τ_{-1}. To choose these values we proceeded as follows:

First, the training set was randomly split into two halves, one half used for training and the other used for testing. Sec-

Table 5: (Submitted Runs) Routing results, PLAUM algorithm. Macro-Average Precision.

Run	TOPICS	MAP	MAP($> .1$)
msPUMb	R101-151	0.355	.368 (#48)
msPUMs	R101-150	0.239	.348 (#34)
msPUMb	R151-200	≈ 0	-
msPUMs	R151-200	≈ 0	-

Table 6: (Post-Submission) Effects of τ and Model Selection (see text for details). Macro-Average Precision for all topics (Test) and for topics R101-150 (Train/Test[50]).

Model	Test	Test[50]	Train[50]
PLA	0.211	0.376	0.4801
PLAUM (+1,0)	0.219	0.385	0.513
PLAUM(*)	0.224	0.403	0.54

ond, the 100 models corresponding to the 100 topics were trained independently for $\tau_{+1} \in \{0, 1, 10, 100\}$ and $\tau_{-1} \in \{-10, -1, 0, 1\}$, leading to 16 different runs per topic. This procedure was repeated 5 times, choosing a different random train/test split every time, and performance on different splits was averaged. This resulted in an average precision reading per topic and per (τ_{+1}, τ_{-1}) setting. Finally, for each topic the best (τ_{+1}, τ_{-1}) parameters were selected and used to train the final model over the entire training set.

The training algorithm was run on a 2.5GHz CPU machine with 500Mb of memory. Data was accessed from a SQL server over a 100Mhz Ethernet network. The entire model selection procedure for the 100 topics and 5 splits runs under 5 hours. We believe that code properly optimised for speed could finalise this task under one hour.

4.4 Results

Due to time and resource limitations we restricted our preliminary experiments to the Reuters-21587 routing task, we have not performed any TREC runs besides those submitted.

Two runs were submitted, varying only in the size of feature set used (as discussed in section 4.2), very large for *msPUMb* and small for *msPUMs*. Results are summarised in 4.4.

The large feature set model msPUMb greatly outperformed the small feature set model on average. This is not surprising, especially when we consider i) how little pre-processing was done with the documents, and ii) the simplicity of the term selection procedure. Nevertheless, for a number of topics the small feature set was better than or similar to the larger feature set. On the left-most column of Table 4.4 we show macro average precision when we average only over the topics obtaining more than 0.1 average precision (we indicated in parenthesis the number of these topics). This figure is very close for both systems, indicating that msPUMs is in fact performing similarly to msPMs for many topics, but it completely underperforms for others. If we could detect such topics at learning time we could adapt the size of the feature sets to the nature of the topics. We are currently working on this problem.

Table 7: (Post-Submission) Some figures of merit of the PLA and the selected PLAUM(*), averaged over all 100 topics.

	PLA	PLAUM(*)
Average Precision	.211	.224
Non-Zero Weights	1179	2236
Epochs	3.6	13.1
Updates	17.5	77.8
Selection time	-	1.87 s.
Train time	.22 s.	-
Train+Test+Submit time	45s.	45s.

There are many algorithms for feature selection and projection which we could have used. However, it has been observed empirically by several authors that using linear classifiers for text seems to benefit from the maximum number of features available. In the absence of space, memory or computational time limitations, we did away with feature selection methods. However, in real operational settings the situation is very different. As one increases the number of features (or similarly if the sparsity of a classifier decreases), the number of potentially relevant documents that need to be scored for each topic increases rapidly. This is very dangerous for systems that must filter simultaneously a large number of topics and documents. Is it then justified to use $10\,000$ features if 80% of the performance can be obtained using only 50 features? This difficult issue is not addressed by the present TREC evaluation measures.

In tables 4.4 and 4.4 we present some results to demonstrate the superiority of the PLAUM algorithm with respect to PLA and the interest in running a model selection procedure such as the one outlined in this paper. For these comparisons we consider only the msPUMb model. We note that these results are better than those submitted originally: after submission we discovered an error in our data normalisation procedure; after correcting it the performance of all models was increased.

In table 4.4 we compare macro-average precision performance (for all topics and for only the first 50) of the original PLA algorithm, a simple PLAUM model with $(\tau_{+1} = +1, , \tau_{-1} = 0)$, and the PLAUM model obtained using the model selection procedure discussed in 4.3. We observe that the original PLA algorithm yields very good performance already and that enforcing some positive margin (i.e. $\tau_{+1} = +1$) increases this performance further. Nevertheless, the best results are obtained when the τs are selected for each topic.

In table 4.4 we compare several figures of merit of the original PLA and our PLAUM(*) model. As expected, learning the PLAUM(*) model requires more updates and more epochs, but its sparsity is not greatly reduced and the resulting training and testing times are perfectly reasonable. In fact, once the model selection step is completed, the difference in training time is negligible compared to IO and scoring time.

5 Conclusions

The performance of the basic Okapi filtering system, tuned for OHSUMED data but run on this year's Reuters task, is fair but not outstanding. The problem of estimating the total number of relevant documents in the entire collection, which is necessary for optimising the FBeta measure, has not been investigated further since last year; it may be one reason why the FBeta-optimised runs performed worse on FBeta than the utility-optimised runs.

The PAUM method for routing appears promising. It could be applied to batch filtering (we have not yet done so); but as with many such machine learning methods, it presents problems if we want to apply it to adaptive filtering. This remains a challenge.

Our performance (with two very different methods on two different tasks) on the intersection topics was extremely poor. This may be because they are simply more difficult, but we suspect that the intersection method is not a very good way to define sufficiently coherent topics.

References

[1] Thore Graepel, Ralf Herbrich, and Robert C. Williamson. From margin to sparsity. In *Advances in Neural Information Processing Systems 13*, pages 210–216, 2001.

[2] W. Krauth and M. Mézard. Learning algorithms with optimal stability in neural networks. *Journal of Physics A*, 20:745–752, 1987.

[3] David Lewis. Applying support vector machines to the trec-2001 batch filtering and routing tasks. In *Text Retrieval Conference (TREC-10)*, pages 286–292, 2001.

[4] Yaoyong Li, Hugo Zaragoza, Ralf Herbrich, John Shawe-Taylor, and Jaz Kandola. The perceptron algorithm with uneven margins. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Nineteenth International Conference on Machine Learning, ICML'2002*, Cambridge, MA, 2002. MIT Press.

[5] S E Robertson et al. Okapi at TREC–3. In D K Harman, editor, *Overview of the Third Text REtrieval Conference (TREC–3)*, pages 109–126. Gaithersburg, MD: NIST, 1995. NIST Special Publication 500-225.

[6] S E Robertson and S Walker. Okapi/Keenbow at TREC–8. In E M Voorhees and D K Harman, editors, *The Eighth Text REtrieval Conference (TREC–8)*, pages 151–162. Gaithersburg, MD: NIST, 2000. NIST Special Publication 500-246.

[7] S E Robertson and S Walker. Microsoft Cambridge at TREC-9: Filtering track. In E M Voorhees and D K Harman, editors, *The Ninth Text REtrieval Conference (TREC–9)*, pages 361–368. Gaithersburg, MD: NIST, 2001. NIST Special Publication 500-249.

[8] S E Robertson, S Walker, and H Zaragoza. Microsoft Cambridge at TREC-10: Filtering and web tracks. In E M Voorhees and D K Harman, editors, *The Tenth Text REtrieval Conference, TREC 2001*, pages 378–383. Gaithersburg, MD: NIST, 2002. NIST Special Publication 500-250.

[9] K Sparck Jones, S Walker, and S E Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management*, 36:779–808 (Part 1) and 809–840 (Part 2), 2000.