A Generative Model for Filtering Thresholds

Yi Zhang Jamie Callan

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15232, USA
{yiz,callan}@cs.cmu.edu

ABSTRACT

This paper presents a generative model of score distribution, focused on the case of information filtering, where sampling of training data is not random. Parameters of the model were estimated using the Maximum Likelihood Principle, conjugate priors, and conjugate gradient descent. Experiments on TREC8 and TREC9 Filtering Track datasets are reported. Our method obtained significant improvements compared to a baseline.

1. INTRODUCTION

Given an initial description of a user information need, a filtering system sifts through a stream of information and delivers documents to a user. While filtering, a user may provide a relevance judgment for each document read. An *adaptive* filtering system can learn from user feedback to improve filtering effectiveness. In order to achieve this goal, the system needs to:

(1) Learn corpus statistics, such as idf for each word;

(2) Learn user profiles, such as adding or deleting terms and adjusting term weights; and

(3) Learn delivery thresholds.

The first problem is comparatively easy. Most of the previous research on information filtering is focused on the second problem. Different variations of an incremental Rocchio algorithm [1][2][3][4][5][6][19][20][21] and other machine learning methods [6][7][8][17][18] have been tested for profile updating while filtering; many have been successful.

Threshold setting has received less attention from the research community, although heuristic measures and regression method [6][15][19][20][21] have been tried. Arampatzis, et. al, proposed a method assuming a Gaussian distribution for the scores of relevant documents and an exponential distribution for the non-relevant documents [2]. However their parameter estimation was biased, because it did not take into consideration the sample bias that occurs during filtering.

2. A GENERATIVE MODEL FOR FILTERING THRESHOLDS

2.1 Generative Model of Score Distribution

Several researchers use a Gaussian distribution for the scores of relevant documents and an exponential distribution for the top ranking non-relevant documents [2][12].

For example:

$$P(score \mid R = r) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(score-u)^2}{2\sigma^2}}$$

$$P(score \mid R = nr) = \lambda e^{-\lambda(x-c)}$$

where u : mean of Gaussian;

σ : variance of Gaussian;

$1/\lambda$: variance of exponential;

c : minimum score for a non-relevant document.

Figures 1 and 2 illustrate how these models fit TREC9 Filtering Track data for OHSU topic 3.

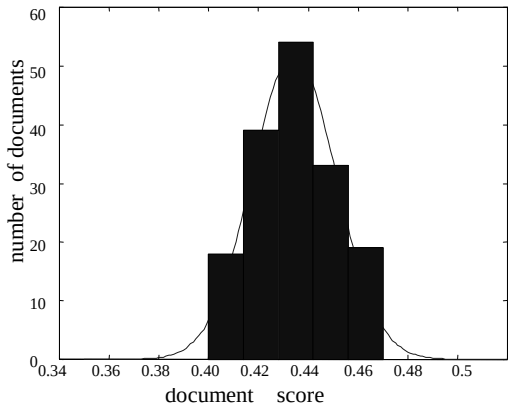

Figure 1. Relevant document scores: OHSU topic 3.

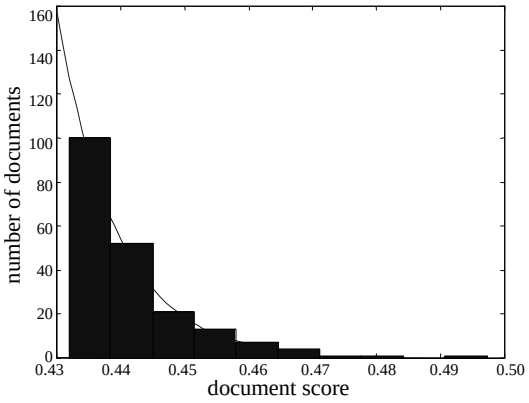

Figure 2. Non-relevant document scores: OHSU topic 3.

Based on this generative model for score distributions, we can calculate the probability of a document being relevant given its score:

$$P(r \mid score) = \frac{P(score \mid r)P(r)}{P(score)} = \frac{P(score \mid r)P(r)}{P(score \mid r)P(r) + P(score \mid nr)P(nr)}$$

$$= \frac{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(score-u)^2}{2\sigma^2}} * p}{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(score-u)^2}{2\sigma^2}} * p + \lambda e^{-\lambda(score-c)} * (1-p)}$$

$$= \frac{1}{1 + \frac{(1-p) * \sqrt{2\pi}\sigma}{p} \lambda e^{\lambda(score-c) + \frac{(score-u)^2}{2\sigma^2}}} = \frac{1}{1 + e^{a+b*score+c*score^2}}$$

Where p : the ratio of relevant documents to all documents.

We have two options: either train a generative model and find the parameters (u, σ, λ, p), or train a discriminative model and find the parameters (a, b, c). Other researchers indicate that a generative model borrows strength from the marginal density and uses training data more efficiently, even when the goal is discriminative [21]. Since our confidence in the model correctness is high, and also because we have little training data during filtering, we decided to go with generative model.

2.2 Non-Random Sampling

Given a set of scores, if we estimate the normal distribution and exponential distribution by calculating the mean and variance over training data [2], the results are biased, because the sample data is restricted to scores above the threshold. We expect that the mean of the sample scores is higher than the real mean.

For TREC9 topic 3, if we get relevance information for all relevant documents, the mean and variance of a Gaussian distribution are (0.4343, 0.0169). In the case of information filtering, if we have a fixed dissemination threshold θ =0.4435, the mean and variance calculated using the biased method proposed by Arampatzis, et. al. [2] are (0.4551, 0.007) (Figure 3).

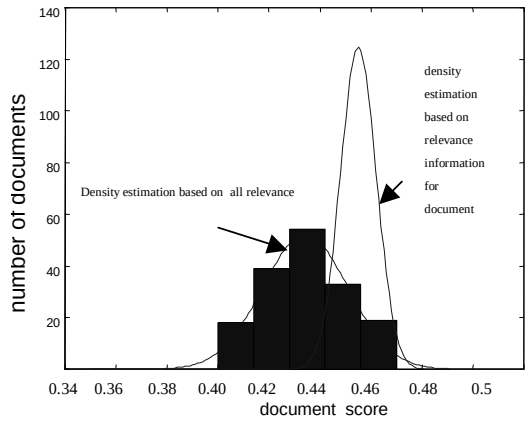

Figure 3. Estimation of parameters for relevant document scores: Topic 3

In order to get an unbiased estimate of the distribution parameters, we must take into consideration the sampling constraint, which is the dissemination threshold. And in the real world of adaptive filtering, the threshold is changing over time, so the problem becomes more interesting.

2.3 Unbiased Estimation Based on the Maximum Likelihood Principle

Maximum Likelihood Estimation, an unbiased parameter estimation method, can be used to solve this problem.

At a certain point in the filtering process, the filtering system has already delivered N documents to a user and relevance judgments for each document are provided by the user. We can treat these documents as training data. For the i th delivered document with user feedback, let's represent it with a triple $(\boldsymbol{R_i, Score_i, \theta_i})$, where: R_i : The user feedback of the document.

$$R_i = \begin{cases} r & \text{for relevant document} \\ nr & \text{for non - relevant document} \end{cases}$$

$\boldsymbol{Score_i}$: The score of this document;

θ_i : The threshold of the profile when the document was delivered.

In order to describe the density distribution of the scores, 4 parameters are necessary: (u, σ, λ, p), where p is the expected ratio of relevant documents in the whole corpus according to the model. Since the exponential model only fits the top non-relevant scores, p does not represent the ratio in the real corpus.

Given the observed training data, which are represented by a set of triples $D = \{(R_i, Score_i, \theta_i), i = 1 to N\}$, according to Bayes theorem, the most probable value of $\mathrm{H} = (u, \sigma, \lambda, p)$ is:

$$\mathrm{H}^* = \arg\max_{\mathrm{H}} P(\mathrm{H} \mid D) = \arg\max_{h} \frac{P(D \mid \mathrm{H})P(\mathrm{H})}{P(D)} \quad (1)$$

For simplicity, we first assume that there is no prior knowledge of the distribution of H and treat the prior probability of $P(\mathrm{H})$ as uniform. (We will revisit this and remove the assumption in Section 3.3.) Because $P(D)$ is a constant independent of H, we can drop it. Thus the most probable H is the one that maximizes the likelihood of the training data:

$$\begin{aligned} &(u^*, \sigma^*, \lambda^*, p^*) \\ &= \arg\max_{(u,\sigma,\lambda,p)} P(D \mid (u, \sigma, \lambda, p)) \\ &= \arg\max_{(u,\sigma,\lambda,p)} \prod_{i=1}^{N} P(D_i \mid \mathrm{H}) \\ &= \arg\max_{(u,\sigma,\lambda,p)} \sum_{i=1}^{N} \log(P(D_i \mid \mathrm{H}) \\ &= \arg\max_{(u,\sigma,\lambda,p)} \sum_{i=1}^{N} \log(P(Score = Score_i, R_i \mid \mathrm{H}, Score > \theta_i) \end{aligned} \quad (2)$$

The second step is due to the assumption that each document is independent; the third step is due to the fact that maximizing a function is equivalent to maximizing its logarithm; and the last step indicates the sampling constraints of training data

For each item inside the sum operation of formula (2), we have:

$$
\begin{aligned}
&P(Score = Score_i, R_i \mid \mathrm{H}, Score > \theta_i) \\
&= \frac{P(Score = Score_i, Score > \theta_i, R_i \mid \mathrm{H})}{P(Score > \theta_i \mid \mathrm{H})} \\
&= \frac{P(Score = Score_i, Score > \theta_i \mid R_i, \mathrm{H}) P(R_i \mid \mathrm{H})}{P(Score > \theta_i \mid \mathrm{H})} \\
&= \frac{P(Score = Score_i \mid R_i, \mathrm{H}) P(R_i \mid \mathrm{H})}{P(Score > \theta_i \mid \mathrm{H})}
\end{aligned} \tag{3}
$$

The last step is due to the fact that all training documents must have a score higher than the threshold.

If we use $g(u, \sigma, \lambda, p, \theta_i)$ to represent the probability of a document getting a score above threshold θ_i, we have:

$$
\begin{aligned}
&g(u, \sigma, \lambda, p, \theta_i) = P(Score > \theta_i \mid \mathrm{H}) \\
&= P(R = r \mid \mathrm{H}) \cdot P(Score > \theta_i \mid R = r, \mathrm{H}) \\
&\quad + P(R = nr \mid \mathrm{H}) \cdot P(Score > \theta_i \mid R = nr, \mathrm{H}) \\
&= p \cdot \int_{\theta_i}^{+\infty} P(Score = x \mid R_i = r) dx \\
&+ (1-p) \int_{\theta_i}^{+\infty} P(Score = x \mid R_i = nr) dx \\
&= p \int_{\theta_i}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-u)^2}{2\sigma^2}} dx + (1-p) \int_{\theta_i}^{+\infty} \lambda e^{-\lambda * (x-c)} dx \\
&= p \int_{\theta_i}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-u)^2}{2\sigma^2}} dx + (1-p) e^{-\lambda(\theta_i - c)}
\end{aligned} \tag{4}
$$

From Equation (2), (3) and (4), we can finally get:

$$
(u^*, \sigma^*, \lambda^*, p^*) = \arg\max_{(u, \sigma, \lambda, p)} \sum_{i=1}^{N} LP_i \tag{5}
$$

where for relevant documents:

$$
LP_i = -\frac{(Score_i - u)^2}{2\sigma^2} + \ln(p / (\sigma \cdot g(u, \sigma, \lambda, p, \theta_i)))
$$

For non-relevant documents:

$$
LP_i = -\lambda(Score_i - c) + ln((1-p)\lambda / g(u, \sigma, \lambda, p, \theta_i)))
$$

2.4 Smoothing: Conjugate Prior

In Section 2.3, for simplicity we set the prior probability of $P(\mathrm{H})$ as uniform. But for the real filtering task, especially at the early stage of filtering with only a very small number of samples, this may cause some problem. For example, if only non-relevant documents have been delivered, estimation of p will be 0 without a prior. If all the relevant documents have the same score, variance will be 0. So we introduced conjugate prior of parameters for smoothing, which can solve all these problems. In our experiments, we set the prior of p as a beta distribution[1]: $p^{\varepsilon_1} \cdot (1-p)^{\varepsilon_2}$, which is equal to adding ε_1 relevant documents and ε_2 none relevant documents sampled randomly for smoothing. The prior of σ is set to be $\exp(-\nu^2 / (2\sigma^2))$, which is equal to adding ν^2 to the sum of the square of the variance of relevant documents[2]. And the prior needn't to be very accurate, because as the number of sample data increases, the influence of the prior decreases. We set $\varepsilon_1 = 0.001, \varepsilon_2 = 0.001, \nu = 0.005$ in our experiment.

[1] Because beta distribution is the conjugate prior for binomial distribution.

2.5 Parameter Optimization Using Conjugate Gradient Descent

There is no closed form solution for Equation 7, so numerical method need to be used. In our experiments, we used conjugate gradient (CG) methods in multi-dimensions to solve this problem.

A sketch of the CG algorithm is shown in Figure 4. At each step, x is the approximate solution, and q is the search direction. x is improved by searching for a better solution along the direction q in each iteration, yielding an improved solution. More detailed description of conjugate gradient method are available in [10][22][23].

Figure 4. Basic idea of the CG algorithm.

x=initial guess for the minimum
q= negative of gradient at x (q : search direction)
do {
    x= the minimal point along direction h
    q= a linear combination of new gradient and old q
} until convergence

3. EXPERIMENTAL METHODOLOGY

3.1 Evaluation Measure

One commonly used evaluation measure for a filtering system is linear utility, which assigns a positive worth or negative cost to each element in the category [14]:

	Relevant	Non-relevant
Delivered	R^+ / A	N^+ / B
Not Delivered	R^- / C	N^- / D

$$
Utility = A \cdot R^+ + B \cdot N^+ + C \cdot R^- + D \cdot N^- \tag{8}
$$

The variables R^+, N^+, R^- and N^- are the number of documents that fall into the corresponding category and A, B, C and D are the gain or cost associated with corresponding category. Filtering according to a linear utility function (8) is equivalent to setting the threshold at θ^*, where:

$$
\frac{C-A}{B-D} \cdot \frac{p}{1-p} \cdot P(score = \theta^* \mid R_i = r) = P(score = \theta^* \mid R_i = nr) \tag{9}
$$

If we let (A, B, C D)=(2, 1, 0, 0), the utility becomes[3]:

$$
T9U' = 2 \cdot R^+ - N^+ \tag{10}
$$

[2] This is a special case of inverse gamma distribution, which is the conjugate prior for variance of normal distribution.

[3] The original T9U measure used in TREC9 is $T9U = Max(2R^+ - N^+, MinU)$, where MinU=-100 for OHSU topics or –400 for Mesh topics.

This is the measure we used in our experiments. The corresponding delivery rule is: deliver if: $P(R = r \mid score) > 0.33$ (11)

3.2 Experimental Setup

Two different text corpora were used to test our algorithm: the OHSUMED dataset and the FT dataset. The OHSUMED data was used in the TREC9 Filtering Track [14]. It is a collection 348566 articles from the US National Library of Medicine's bibliographic 1987-1991 database [11]. 63 OHSUMED queries and 500 MeSH headings were used to simulate user profiles. The average number of relevant articles in the testing data is 51 for OHSUMED topics and 249 for Mesh headings.

The FT data was used in the TREC8 Filtering Track [6]. It is a collection of 210158 articles from the 1991 to 1994 Financial Times. TREC topics 351-400 were used to simulate user profiles. The average number of relevant articles in the testing data is 36.

For each topic, the filtering system creates an initial filtering profile and sets the initial threshold to allow the top $\delta = 3$ documents in the training dataset to pass. For simplicity, user profile term weights are not updated while filtering.

Because the first two relevant documents given to the system were not sampled under these constraints, their probabilities are simply $P(d_i \mid R_i = r)$, and the corresponding part of Equation 7 was changed to:

$$LP_i = -\frac{(Score_i - u)^2}{2\sigma^2} - \ln(\sigma) .$$

For each query set, 4 runs were carried out. The first one used the parameter estimation method described in [2]. The third one used our maximum likelihood estimation. Both runs will stop delivering documents when Δ is negative (Equation 12), so a minimum delivery ratio was introduced to avoid this problem. If a profile has not achieved the minimum delivery ratio (set to deliver at least 10 documents in the whole time period), its threshold is decreased automatically. This corresponds to the second and fourth runs. For our algorithm (ML plus minimum delivery ratio), it takes only about 21 minutes for the whole process of 63 OHSU topics on 4 years of OHSUMED data.

4. EXPERIMENTAL RESULTS

According to the experiment results (Table 1, Figure 6) for the OHSUMED dataset, none of the algorithms work well without using a minimum delivery ratio to decrease the threshold when it is set too high.

On the OHSUMED dataset, for both OHSU topics and MESH topics, using maximum likelihood estimation plus the minimum delivery ratio achieved the best results. Although profile updating is disabled while filtering, all of the runs on this dataset get a positive average utility. The OHSU topics result for run 4 is above average compared with other filtering systems in the TREC9 adaptive filtering track. This indicates how efficient the threshold setting algorithm is, considering all of the other filtering systems are updating profiles while filtering.

On the FT data, the performance of the four methods is almost the same (Table 2). One difference between the FT dataset and the OHSUMED dataset is the average number of relevant documents in the testing set per profile (Section 3.2). In the FT data set, most of the user profiles are not good profiles, which means it is almost impossible to get a good threshold that achieves a positive utility without profile updating. In this case, the ML method does not improve the performance much.

Table 1: Utility of each filtering runs: OHSUMED dataset

		Run 1	Run 2	Run 3	Run 4
		Method in [2]	Method in [2] + min. delivery Ratio	ML	ML + min. deliver y ratio
OHSU topics	T9U utility	1.84	**3.25**	2.7	**8.17**
	Avg. docs. delivered per profile	3.83	9.65	5.73	18.40
	Precision	0.37	0.29	0.36	0.32
	Recall	0.036	0.080	0.052	0.137
MESH topics	T9U utility	1.89	**4.28**	2.44	**13.10**
	Avg. docs. delivered per profile	3.51	11.82	6.22	27.91
	Precision	0.42	0.39	0.40	0.34
	Recall	0.018	0.046	0.025	0.068

Table 2: Utility of each filteirng runs: FT dataset.

		Run 1	Run 2	Run 3	Run 4
		Method in [2]	Method in [2] + min. delivery Ratio	ML	ML + min. delivery ratio
TREC topics	T9U utility	1.44	**-0.209**	0.65	**0.84**
	Avg. docs. delivered per profile	9.58	10.44	9.05	12.27
	Precision	0.20	0.17	0.22	0.26
	Recall	0.161	0.167	0.15	0.193

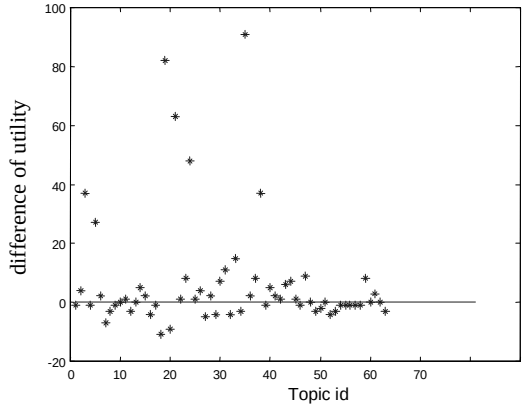

Figure 5. Utility differences on OHSU topics: ML-Run2. For most of the topics (points above the horizontal line), ML (run 4) has a higher utility than run 2.

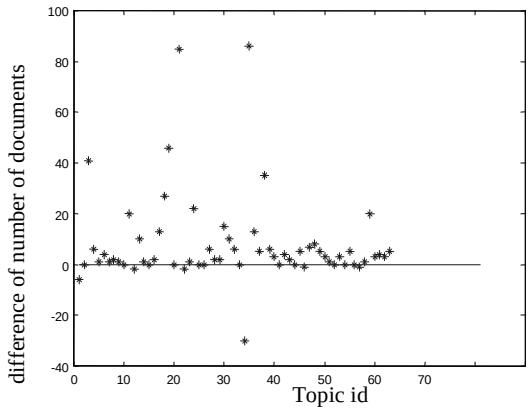

Figure 6. Number of docs delivered difference on OHSU topics: ML-Run 2. For most of the topics, ML (run 4) delivered more documents than run 2.

According to the results on the OHSUMED dataset, we found that the average number of documents delivered to a profile for run 2 is less than run 4 (Figure 5). And for run 2, it is around the required number set by minimum delivery ratio. We believe that the mean of Gaussian distribution based on run 2 is too high compared to the actual mean (Figure 3), thus the threshold is set too high.

5. CONCLUSION AND FUTURE WORK

Based on a generative model of score distribution, using a Gaussian distribution for the scores of relevant documents and an exponential distribution for the scores of non-relevant documents, we have developed an effective algorithm for threshold setting while filtering that maximizes a given linear utility measure.

Due to the fact that only relevance judgments of delivered documents are available, we proposed an unbiased algorithm based on the maximum likelihood principle to jointly estimate the parameters of the two density distributions and the ratio of the relevant document from user feedback about delivered documents sampled under the constraint of a changing threshold. We believe this is the first paper that solves the sample bias problem for information filtering. Our new method obtained significant improvement on the TREC-9 Filtering task.

In our experiments, the profile is not updated while filtering. Although the effectiveness of the system is very competitive, we believe combining threshold updating with profile updating will achieve better accuracy. One difficulty of combining our algorithm with profile updating is adjusting training data $D = \{(R_i, Score_i, \theta_i), i = 1 to N\}$, especially θ_i based on new profile terms and weights. Future work can be focused on this problem.

6. ACKNOWLEDGMENTS

This material is based on work supported by Air Force Research Laboratory contract F30602-98-C-0110. Any opinions, findings, conclusions or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsors.

7. REFERENCES

[1] J. Allan. 1996. Incremental Relevance Feedback for Information Filtering. In *Proceedings of SIGIR 1996*

[2] A. Arampatzis, J. Beney, C.H.A. Koster, and T.P. van der Weide, (In Press). KUN on the TREC-9 Filtering Track: Incrementally, Decay, and Threshold Optimization for Adaptive Filtering Systems. In *Proceeding of ninth Text Re Retrieval Conference (TREC-9)*, NIST.

[3] J. Broglio, J.P. Callan, W.B. Croft, and D.W. Nachbar, 1995. Document retrieval and routing using the INQUERY system. In *Proceeding of Third Text Retrieval Conference (TREC-3)*, NIST.

[4] J. Callan. 1996. Document Filtering With Inference Networks. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[5] J. Callan. 1998. Learning while Filtering. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*

[6] D A. Hull, S. E. Robertson. 1999. The TREC-8 Filtering Track Final Report. In *Proceeding of eighth Text Retrieval Conference (TREC-8)*, NIST.

[7] Y.H. Kim, S.Y. Hahn, and B.T. Zhang. Text Filtering by Boosting Naive Bayes Classifiers. In *Proceedings of SIGIR 2000*, ACM Press.

[8] R.D. Lyer, D.D. Lewis, R.E. Schapire, Y. Singer, A. Singhal. Boosting for Document Routing. In P*roceedings of CIKM 2000*, ACM Press

[9] M. F. Porter, 1980. An algorithm for suffix stripping.

[10] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. Numerical Recipes in C. Cambridge University Press, 420-425.

[11] W. Hersh, C. Buckley, T. J. Leone and D. Hickam 1994. OHSUMED: an interactive retrieval evaluation and new large test collection for research. In *Proceedings of SIGIR 94, ACM Press*

[12] R. Manmatha, T. Rath, F. Feng 2001 (In Press). Modeling Score Distributions for Combining the Outputs of Search Engines. To appear in the Proceedings of ACM SIGIR 01 conference, New Orleans, LA.

[13] J. J. Rocchio. 1971. Relevance feedback in information retrieval in The SMART Retrieval System-Experiments in Automatic Document Processing, Page 313-323. Prentice Hall Inc.

[14] S. E. Robertson, D. A. Hull 2000. Guidelines for The TREC-9 Filtering Track.

[15] S.E. Robertson, S. Walker. Microsoft Cambridge at TREC-9: Filtering track. In *Proceeding of ninth Text Re Retrieval Conference (TREC-9)*, NIST.

[16] S. E. Robertson, S. Walker, M. M. Beaulieu, and M. Gatford, A. Payne, 1995. A. Okapi at TREC-4. In

Proceeding of Fourth Text Retrieval Conference (TREC-4), NIST.

[17] R.E. Schapire, Y. Singer, and A. Singhal. Boosting and Rocchio Applied to Text Filtering. In *Proceedings of SIGIR 98, ACM Press*

[18] T.Ault, Y. Yang. kNN at TREC-9: A Failure Analysis. In *Proceeding of ninth Text Re Retrieval Conference (TREC-9)*, NIST.

[19] C. Zhai, P. Jansen, E. Stoica.1998. Threshold Calibration in CLARIT Adaptive Filtering. In *Proceeding of seventh Text Retrieval Conference (TREC-7)*, NIST.

[20] C. Zhai, P. Jansen, N. Roma, E. Stoica, D.A. Evans 1999. Optimization in C LARIT Adaptive Filtering. In *Proceeding of eighth Text Retrieval Conference (TREC-8)*, NIST.

[21] Y. Dan Rubinstein, T. Hastie 1997. Discriminative vs Informative Learning. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining.*

[22] http://wol.ra.phy.cam.ac.uk/mackay/c/macopt.html

[23] http://www.ccr.buffalo.edu/class-notes/hpc2-00/odes/node4.html